BIT-LET

öttlet SZÁMÍTÁSTECHNIKAI MELLÉKLET · 1987 DECEMBER 51

BÚCSÚZUNK!

A BIT-LET több mint négy éve kezdte meg „működését". Az 51. szám – ez, amit most ön a kezében tart – mérföldkő lapocskánk történelmében. Ezzel a számmal ugyanis megszűnünk.

VAN JÖVŐNK?

Remélhetőleg ez a megszűnés egyben valaminek a kezdetét is jelenti. A tervek, elképzelések szerint ugyanis csak most lesz igazán BIT-LET. Jövőre ugyanis önálló lapként szeretnénk megjelenni a piacon. Az első időkben aligha leszünk havilap. Célunk persze nyilván ez.

ELŐRE!

Igyekszünk majd, hiszen már magát a tényt is előrelépésnek tekintjük, hogy kiléphetünk anyalapunk kebeléből. De a lap profiljában is sokféle változást, bővülést tervezünk. Nyitni szeretnénk a profik felé is. Mert úgy gondoljuk, hogy amit nem tehettünk meg eddig havi 16 oldalon, azt megtehetjük 32–40 oldalon.

SOSEM HÁTRA!

Ez a jelszó azt is jelenti, hogy ami jó volt a 16 oldalas BIT-LET-ben, azt természetesen megtartjuk. Sőt, bővíteni kívánjuk sikeres rovatainkat, továbbra is nagyszájúak akarunk maradni, továbbra is célunk az amatőrök érdekeit szolgálni.

KÖSZÖNET

Azoknak, akik lapunk eddigi pályafutása során bizalmat szavaztak nekünk, s törzsolvasóinkká váltak, őket szeretnénk megtartani a jövőben is. Köszönet azoknak, akik programokat küldtek lapunknak, akik cikkekkel járultak hozzá a 16 oldal elkészítéséhez. Munkájukra az új lap létrejötte esetén is igényt tartunk. Kérjük tehát, hogy ne veszítsenek szem elől bennünket.

JOGFOLYTONOSSÁG

Szeretnénk mindazok türelmét kérni, akiknek cikke, programja vár nálunk közlésre. Az új lap – a jelenlegi folytatása – szerkesztése során mindazokat a közlésre érdemes anyagokat szeretnénk felhasználni, amelyek ma fiókunkban vannak.

BIZONYTALANSÁG

Persze a dolgok természetéből következik, hogy ma még nem tudjuk pontosan megmondani, hogy mikor és hány oldalon, valamint, hogy milyen áron jelenik majd meg az új lap. Mindazoknak, akik előzetes információra tartanak igényt, fölhívjuk a figyelmét a 30. oldalon található kis értesítő kártyánkra. Ígérjük, hogy nem felejtkezünk el az előzetes értesítésről.

VÉGÜL

Végül búcsúzik a szerkesztő, boldog új évet kíván minden BIT-LET olvasónak, s első ízben jelzi, hogy a jövő havi megjelenés dátumát hiába keresik a melléklet 16. oldalán.

BELÜLRŐL

HÍROLDAL



...LIUMNIOBÁT

...y kutatási program keretében a zürichi ...úszaki egyetem szilárdtest-fizikai laborató...umában sikerrel növesztenek káliumniobát kristályokat. Ezeknek a nagy értékű mester...séges kristályoknak az előállítása – kutatók véleménye szerint – mérföldkő az optikai rendszerek fejlesztésében. Ezek segítségé...vel a számítógép elektronikus vezérlésű áram...köreit optikaiakkal lehet majd felváltani. Zürichben állítólag sikerült a káliumniobát kristályokat a világon eddig még el nem ért tisztaságban előállítani, és ezeknek nagyon jók az optoelektronikai tulajdonságaik.

TÖRÖLHETŐ LEMEZ

A Sony eredetileg informatikai adattároló céljára fejlesztette ki törölhető és újraírható optikai lemezét. A 13 cm átmérőjű lemez oldalanként 300 millió byte-ot tárol – ez 200 000 gépelt oldallal egyenértékű – szemben a kompaktlemez egymillió byte-jával. A lemez nem jelent konkurenciát a videoszalagnak, a mozgókép egyébként is hatalmas mennyiségű információ tárolását igényli. Komoly versenytársává válhat azonban a kompaktlemeznek.

JAPÁN ROBOTOK

Japánban napjainkban már százezer teljesen automatizált robot működik. A gyárakban az összes szerelési munka 20 százalékát gépek végzik. Ezzel szemben például az Egyesült Államokban csak húszezer robotot foglalkoztatnak, s ezek felét is a gépjárműgyártásban. A Fanuc japán robotgyár ebben az évben már havi ezer robotot gyárt, ezenkívül az egész világon készülő különböző géptípusokhoz különleges szerszámgépeket, s központi vezérlő berendezéseket. A kutató laboratóriumokban pedig olyan gépeket terveznek, amelyek képesek lesznek önállóan dolgozni a tengerfenéken, tüzeknél helyettesítik a tűzoltókat, s az atomerőművekben is dolgozni tudnak.

-CLN

NDK CHIP

Új chip gyártását kezdték meg az NDK-ban. A saját fejlesztésű, 256 kilobit kapacitású integrált áramkörök készítésére ma még csak néhány vezető tőkés ország képes. Az NDK-ban jelenleg 120 ezren dolgoznak a mikroelektronikai alkatrészek, részegységek kutatásában, fejlesztésében és gyártásában.

TELEFONKÖNYV

Másfél évvel ezelőtt döntött a Magyar Posta arról, hogy a korábbi bontás helyett megyénként adja ki a telefonkönyveket. Az első hat megyei telefonkönyv elkészülte után a Posta megállapodást kötött az Idegenforgalmi Propaganda és Kiadó Vállalat TD képviseletével, s ez gyökeres változást hozott a névsorok szerkesztésében, előállításában. A megállapodás alapján ugyanis ez év márciusa óta az IPV TD számítógépe végzi a telefonkönyv tördelését, szedését. A gép körülbelül öt percenként állít össze egy oldalt, amiről lézer másoló készít nyomdakész levonatot. Ezt lefotózzák, és a filmről sokszorosítják a telefonkönyv lapjait. A számítógép folyamatosan ellenőrzi az adatokat és jelez, ha hibára akad. Előnye az új módszernek az is, hogy jóval gyorsabb, mint a hagyományos eljárás: egy öthasábos, komplett könyvoldalt készít annyi idő alatt, amennyi egy hasáb kézi tördeléséhez szükséges. A gyors átfutási időnek köszönhetően a legfrisebb adatváltozások is szerepelnek az új kötetekben.

AUTOMATÁBÓL

Az Egyesült Államokban szoftvert is lehet kapni automatából. A Computerland cég 550 üzletben olyan automatákat állított fel, amelyekből a vevő 1000-nél több programot választhat ki. A programokat optikai lemezes memóriában tárolják. A vevőnek először bemutatják a szoftvert ismertető rövid videofilmet, azután megvizsgálhatja a kiválasztott programot és ha megfelel, készíthet róla hajlékony lemezre másolatot. Ezt egy rövid leírással együtt megkapja. Az eredeti kézikönyv néhány nap múlva postán érkezik.

SZOLFÉZS

A zenetanítás egyik alaptárgya a szolfézs, alapos elsajátítására viszont a tapasztalatok szerint a zeneiskolákban rendelkezésre álló heti két óra nem elegendő.

Az otthoni és iskolai gyakorlást teszi most hatékonyabbá és könnyebbé az a számítógépes programcsomag, melyet Kalmár Gyula, az Egri Állami Zeneiskola tanára alkotott meg és a Tudományszervezési és Informatikai Intézet forgalmaz. A Commodore 64 és 128 típusú számítógépeken futtatható programok használata nem igényel számítástechnikai jártasságot, ám segítségükkel a tanulók elmélyültebb ismeretséget köthetnek a hangközökkel, hangzatokkal, hangnemekkel, és fejleszthetik dallamírási készségüket is.

VÍVÁS

1987. október 1-jétől a Datacoop lett a Magyar Vívó Szövetség hivatalos támogatója. Ami az információban a legfontosabb és leglényegesebb: a számítástechnikai kisszövetkezet havonta százezer forinttal támogatja a vívószövetséget, s miután a megállapodás öt esztendőre szól, a végösszeg hatmillió forint! A Datacoop elnöke szerint a prosperáló cég számára kitűnő reklámhordozónak tűnik a magyar vívóválogatott, amely az utóbbi időben több jelentős nemzetközi eredményt produkált, a lausanne-i vb-n egyenesen remekelt. Ezért kötötték az 1992. december 31-ig érvényes megállapodást. A szerződés természetesen a vívószövetségre is ró kötelezettségeket. Vállalta, hogy a felnőtt, a junior és a kadett válogatott tagjai – az Olimpia kivételével – valamennyi világversenyen és a hazai nemzetközi viadalokon melegítőjükön és vívóruhájukon feltűnő helyen viselik a Datacoop feliratot. Ez a jelenlegi szabályok szerint három, tízszer-tíz cm-es felület. Mi több, nemcsak a melegítőn és a vívóruhán, hanem a vívózsákon, -táskán, fürdőköpenyen és a trikón is elhelyezik az új szponzor nevét. Abban is megállapodtak, hogy az MVSZ kötelezi a nemzetközi és hazai viadalokon döntőbe jutott, nem válogatott versenyzőket is a Datacoop felirat viselésére.

GYÓGYSZER

A Szovjetunió Gyógyszerészeti Tudományos Kutató Intézetében kidolgozták a Gyógyszer nevű számítógépi rendszert, amelytől az orvosok naprakész információt kaphatnak a forgalomban lévő gyógyszerekről, azok klinikai alkalmazásáról, arról, hogy mely készítmények helyettesíthetik egymást és – összeférhetetlenség miatt – melyek nem adhatók egyidejűleg a betegnek. A rendszer felvilágosítást ad a gyógyszerek fizikokémiai tulajdonságairól, minőségük ellenőrzésének és tárolásuknak a módjáról is. A még az idén üzembe helyezendő rendszer adatállományát rendszeresen kiegészítik majd az újonnan megjelenő bel- és külföldi gyógyszerkészítmények adataival. Az új információs rendszer előnyeit nemcsak a moszkvai egészségügyi intézményekben hasznosíthatják majd, hanem a fővároson kívüli központokban is, ahová az adatokat mágnesszalagon juttatják el.

TRÓFEABÍRÁLAT

A számítástechnika rohamos elterjedése következtében életünk egyre több területén találkozhatunk számítógépekkel. Tavaly az Országos Trófeabíráló Bizottság budapesti központja és a MÉM Erdőrendezési Szolgálat arra az elhatározásra jutott, hogy számítógépet használ a bírálóbizottságok munkájának könnyítésére és pontosítására. A számítógépes bírálat esetében a méretfelvételeket ugyanúgy el kell végezni, mint a hagyományos módon. A méretek itt is törzskönyvbe kerülnek, de a számítógép pillanatok alatt kiszámolja a trófea pontszámát, és az adatokat mágneslemezen tárolja. A mágneslemezes tárolás előnye, hogy sok adatot lehet nagyon kis helyen tárolni, és ki lehet küszöbölni a számolási hibákat, pontatlanságokat is. A számítógép segíti az adatfeldolgozást is. A gép a mágneslemezen tárolja az adatokat a meghatározott program szerint, és azokat gyorsan feldolgozza. Ebben az esetben is sok mechanikus számolást és időt takarít meg. A terveknek megfelelően az OTB mellett a győri, a keszthelyi, és a székesfehérvári bizottságoknál is bevezetik a számítógépes bírálatot. A bíráló- és adatfeldolgozó programok egységesek, így a trófeabíráló bizottságok mágneslemezen tárolt adatait az OTB-nál gyorsan és pontosan lehet feldolgozni. Így az országos adatfeldolgozás is jóval gyorsabban végezhető el, mint a hagyományos módon. Számítógép segít a vadfajok világranglistájának és magyar ranglistájának elkészítésében és azok naprakész vezetésében is. Az OTB tervezi, hogy a számítógép segítségével tájékoztatni tudja a vadásztatókat a vidéki trófeabíráló bizottságoknál bemutatott külföldi és hazai vadászok által elejtett vadfajok trófeáinak adatairól is.

NYELVTANULÁS

Nyolchetes, bentlakásos angol nyelvtanfolyamot indított október közepén a TIT soproni szervezete. A tanfolyam hallgatói hazánkban egy eddig még nem alkalmazott módszerrel tanulják meg az idegen nyelv alapjait. Az újszerűség abból áll, hogy a mikroelektronikát a nyelvtanulás szolgálatába állítják. Valamennyi hallgató kap egy mikroprocesszorral vezérelt oktatókészüléket, amely a légzésritmusból érzékeli használója idegállapotát, s csak akkor indítja el önműködően a magnetofonkazettára rögzített programot, amikor a nyelvtanuló nyugalmi állapotba kerül; a latzítást kellemes fény- és hangjelek továbbításával segíti az NSZK-ból érkező készülék. A tanfolyamon nyelvtanár, orvos és pszichológus is a hallgatók rendelkezésére áll.



ÚJ!

AST PREMIUM/386

Az amerikai AST Research Corporation cég legújabb produkciója az Intel 80386 típusú mikroprocesszorra alapozott, AST Premium/386 névre keresztelt számítógépe. Az első 20 MHz-es, 32 bites számítógép memóriája 2-től 13 Megabyte-ig bővíthető. 40, 90 és 150 Megabyte-os fix lemezegységek kapcsolhatók hozzá. Beépített floppy egysége 5 1/4 inctes, 1,2 Megabyte-os. Kitűnően üzemeltethető MS OS/2 és XENIX operációs rendszerek alatt. Az új számítógép különösen előnyösen alkalmazható CAD/CAM grafikus feladatokra és hálózati rendszerekben.

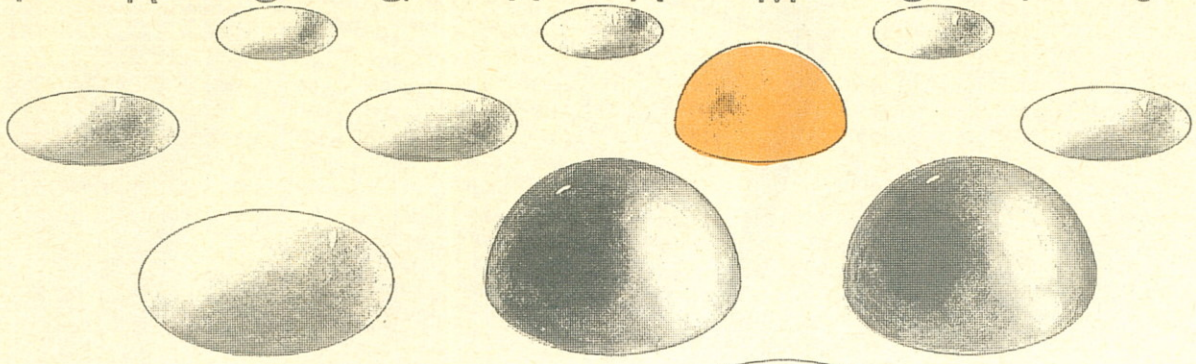
ORVOSKÉPZÉS

A budapesti Semmelweis Orvostudományi Egyetemen az orvostanhallgatók képzésében már rendszeresen alkalmaznak olyan számítástechnikai programokat, amelyek segítségével a medikusok alaposabban elsajátíthatják a tananyagot. A személyi számítógépekre alkalmas programokból pedig huszonötöt készítettek a Biofizikai Intézet munkatársai, s ezek közül többnek az összeállításában a hallgatók is részt vettek.

A programok segítségével tanulják meg a medikusok többek között az EKG-görbék keletkezését, s modellezik a vérkeringést. Megismerkednek az orvostudományban alkalmazott legfontosabb műszerek felépítésével, működésével, valamint a számítógépek gyógyászati felhasználásának lehetőségeivel. A hallgatók a programokat a laboratóriumi gyakorlatokon hetente rendszeresen tanulmányozzák, de ezenkívül önszorgalomból bármikor újból áttekinthetik. Az eddigi tapasztalatok szerint a szemléletes módszerrel a medikusok könnyebben sajátítják el a tananyagot.

R O T A R Y

P R O G R A M · C 1 6 - R A



Az ENTERPRISE-nyerő első fordulójának (Rotary játék) kiértékelése még tart, azonban a kazettán és lemezen beérkezett programokat végignéztük. 3 olyan program érkezett, melyek játszani is tudnak (nem csak adminisztrálni). Ezek szerzői Baracs Tibor (ZX Spetrum), Dévényi Imre (C 16) és Nagy László (C 64). Mindhármójuknak gratulálunk, a programokat köszönjük (a többi kazettát, ill. lemezt beküldő pályázónknak is)

Barács Tibor programja meglehetősen hosszú, s elég hosszú, részletes leírás is tartozik hozzá – ezt az anyagot egy későbbi számunkban szeretnénk közölni.

Nagy László leveléből idézünk néhány sort:

„Bevallom, nem vagyok rendszeres olvasója lapjuknak. Barátaim hívták fel a figyelmemet, hogy a ROTARY valamilyen pályázatukban szerepel. Érzékenyen érintett az információ két oldalról is:

– Egyrészt, a ROTARY, mint mechanikus játék, magyar szabadalommal védett, saját szellemi termékem.

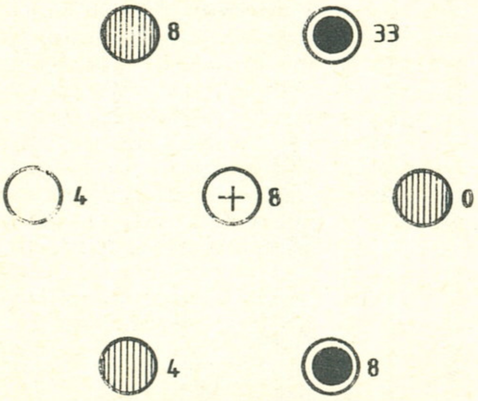
– Másrészt, közel három éve családi és közvetlen baráti környezetemben sikeresen működik a játék C 64-re írt változata.

Néhány szót a játék történetéről: 1981-ben kb. három hónapot fordítottam különböző új típusú logikai játékok kidolgozására, részben kedvtelésből, és bevallom, részben pénzszerzési szándékból Az ebben az időszakban született kb. 15 játékötletből hármat értékeltem találmányi szintűnek, ezekre meg is kaptam a szabadalmi oltalmat.

Termék egyedül a ROTARY-ból lett, 1984 karácsonya előtt. Annak ellenére, hogy a játék ún játékértéke magasnak mondható (intellektuálisan is igényes), piaci megítélése kedvezőtlen volt, így az első széria után, 1985 elején a MÜTEX leállt a gyártással. (A legcsekélyebb reklám nélkül került piacra, magas áron – a csomagolás költségei

```
0 REM *** ROTARY C16-RA / DEVENYI / ***
2 GRAPHIC0,1:CLR:COLOR0,10,5:COLOR4,10,3
4 DATA 0,0,0,1,1,1,1,1,1,0,0,1,1,0,0,0,0
6 DATA 0,0,1,1,0,0,0,0,0,0,1,1,0,0,0,0,0
8 DATA 0,-1,1,0,0,0,0,0,0,-1,0,0,0,0,0,0
10 DATA 0,-1,0,0,0,0,0,0,0,-1,0,0,0,-1
12 DATA -1,-1,-1,-1,0,0,0,0,-1,-1,-1,-1
14 DATA 0,0,0,0,0
16 DIM T%(8,8),X%(8,8),P%(8,8)
18 DIM G$(3),P$(3),SZ$(4)
20 DATA 0,0,0,2,4,8,8,4,0,0,0,2,4,8,0,8
22 DATA 4,0,0,2,4,8,33,33,8,4,0,0,4,8,0
24 DATA 33,0,8,0,0,0,4,8,8,8,8,0,0,0,0,4
26 DATA 4,4,4,0,0,0,0,0,2,2,2,0,0,0,0,0
28 DATA3,5,4,4,3,4,2,5,4,5,4,3
30 DATA3,3,2,4,2,6,2,7,3,6,5,4,5,3
32 DATA6,3,2,3,6,2,5,2,4,2,3,2,3,1
34 DATA4,1,5,1,6,1,7,1,7,2,7,3,6,4,5,5
36 DATA4,6,3,7,1,7,1,6,1,5,1,4,1,3,2,2
38 VOL7
40 CHAR,12,6,"R O T A R Y":PRINT
42 PRINT"▓▓   A JATEKTEREN"; :REM "▓"=CRSR LE
44 PRINT" A GOLYOKAT A KORONG":PRINT
46 PRINT"  LETEVESE UTAN (PL:F-7),";
48 PRINT" A KORONG EL-":PRINT
50 PRINT"  FORGATASAVAL TUDJUK";
52 PRINT" MOZGATNI.":PRINT
53 PRINT"   (PL: JOBBRA 2*60 FOK = J-2)":PRINT
54 PRINT"  RONTAS ESETEN MINDKET BEADASNAL"
55 PRINT:PRINT"  A SZAM HELYETT 0-T KELL IRNI."
56 PRINT:PRINT"   CEL A BELSO ";
58 PRINT"HAROMSZOG ELFOGLALASA!"
60 RESTORE:FORI=0TO8:FORJ=0TO8
62 READ A:T%(I,J)=A
64 X%(I,J)=-2+4*I+2*J:NEXTJ,I
66 FORI=1 TO 7:FORJ=0 TO 8
68 READ A:P%(I,J)=A:NEXTJ,I
70 G$(3)=CHR$(144)+CHR$(113)
72 G$(1)=CHR$(5)+CHR$(113)
74 G$(2)=CHR$(129)+CHR$(122)
76 FORI=1TO39:BL$=BL$+" ":NEXT
78 FORI=1TO11:F$=F$+"/▓":L$=L$+"\▓" :REM "▓,▓"=CRSR LE,FEL
80 E$=E$+"_":K$=K$+"▓ ▓":NEXTI :REM "_"=C< P, "▓,▓"=RVS ON,OFF
82 FORI=1TO7:K1$=K1$+"▓▓▓▓▓" :REM "▓"=CRSR LE, "▓"=C< *
84 K3$=K3$+"▓▓▓▓▓":NEXT I :REM "▓"=CRSR FEL, "▓"=SHIFT £
86 GETKEY B$:PRINT"▓" :REM "▓"=CLEAR
88 FORI=1TO7
90 CHAR,0,2+2*I,CHR$(56-I)+"-":NEXT
92 CHAR,2,12,"| ▓▓"+F$+LEFT$(E$,4)+E$+"▓"+L$+"▓ |" :REM "| "=C< G, " |"=C< M
94 CHAR,2,13,K1$+K$+K$+"▓ ▓"+K3$
96 CHAR,3,17,"A▓/▓▓▓B▓/▓▓▓C▓/▓▓▓D▓/▓▓▓E▓/▓▓▓F▓/▓▓▓G▓/" :REM "▓"=CRSR JOBBRA, "/"
=SHIFT N
98 KO$="o▓▓▓▓▓▓ ▓▓/▓/▓___--‾--__▓\▓\▓▓ ▓▓/▓▓▓/▓▓▓▓▓▓▓▓▓▓▓\▓\▓‾‾--‾‾" :REM "o,\"=SH
IFT W,M
99 REM"__--‾‾"=C< P, SHIFT R,F,C,D,E, C< Y; A 6. "▓" UTAN C< G, A 4. "▓" ELOTT
C< M VAN!
100 K0$="♦▓▓▓▓▓▓ ▓▓ ▓ ▓         ▓ ▓ ▓▓ ▓▓ ▓▓▓ ▓▓▓▓▓▓▓▓▓▓▓ ▓ ▓         " :REM "♦"=SHI
FT Z
102 GOSUB276
104 GOSUB106:GOTO118
106 FORI=0TO8:FORJ=0TO8:REM *** RAJZ ***
108 IFI+J<3 ORI+J>11THEN114
110 B=2*(9-J):A=X%(I,J)
112 CHAR,A,B,G$(T%(I,J)+2)
114 NEXTJ,I
116 RETURN
118 SZ$(0)=" JATEKOS! KORONG HELYE?     "
120 SZ$(3)="▓FEKETE":SZ$(1)="▓FEHER " :REM "▓,▓"=C< 1,2
122 SZ$(2)="FORGATAS IRANYA, HANYSZOR 60 FOK?    "
124 CHAR,2,23,"LEGYEK EN A FEHER JATEKOS? (I/N)"
126 GETKEY B$:IFB$="I"THEN AU=-1:GOTO132
128 IF B$="N"THEN AU=0:GOTO132
130 GOTO126
132 CHAR,0,22,BL$:CHAR,0,23,BL$
134 G=1:LS=0:GY=0:REM** JATEK KEZDES ***
136 IF G=-1 AND(AU)THEN 302
138 LS=LS+1:CHAR,36,1,STR$(LS)
140 CHAR,2,22,SZ$(G+2)+SZ$(0)
142 GETKEY B$:IF B$<"A"OR B$>"G"THEN142
144 X=ASC(B$)-64:CHAR,32,22,B$+"-"
146 GETKEY B$:IF B$<"0"OR B$>"7"THEN146
148 IFB$="0"THENCHAR,32,22,"   ":GOTO142
150 Y=VAL(B$):CHAR,34,22,B$:GOSUB216
152 IF N=0 THEN CHAR,32,22,"???":GOTO142
154 ON G+2 GOSUB288,290,286
156 CHAR,2,23,SZ$(2)
158 GETKEY B$
160 KF=(B$="B")-(B$="J"):IF KF=0 THEN 158
```

meghaladták a tartalom előállításának költségeit stb.)."
Gratulálunk Nagy Lászlónak a szerintünk is igen magas szintű, gondolkodtató, élvezetes és épp ezért jobb sorsra érdemes játékhoz. Nagyon sajnáljuk, hogy ez a játék is a magyar találmányok nagy részének sorsára jutott. Szerintünk a játék ára a mai játékárakhoz viszonyítva nem volt kiugróan magas, valóban a reklám hiánya (és a csomagolás) lehetett a kudarc egyik fő oka, a másik talán a kitartás hiánya (mármint a MÜTEX részéről), ugyanis tapasztalataink szerint a magyar játékpiacon nem ritka jelenség, hogy egy játék csak 2–3 évvel árusításának megkezdése után fut be igazán (ez a késési idő még a Bűvös kockánál is kb. két év volt). Nagy László igen jól játszó programját a Commodore újság szeretné közölni.

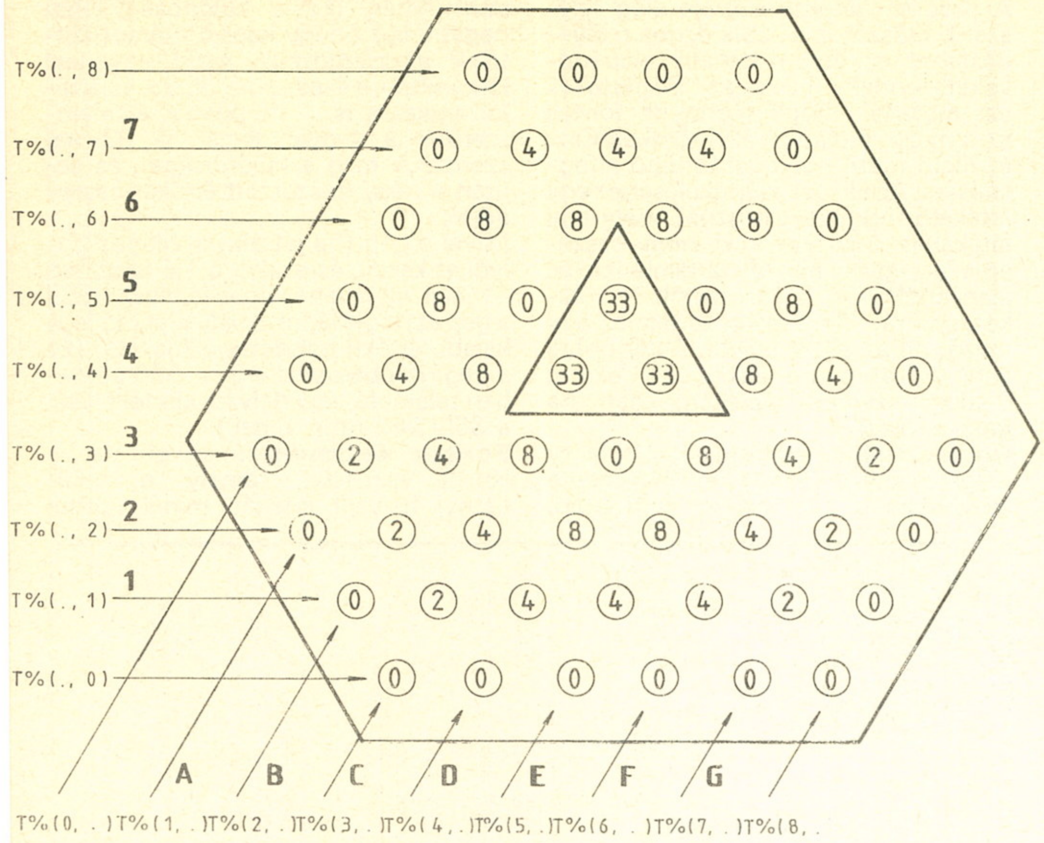


A PROGRAM

Mi most **Dévényi Imre** programját közöljük (kisebb változtatásokkal), mely ugyan egy kicsit gyengébben játszik a másik kettőnél, képernyőképe sem olyan szép, viszont csak fele olyan hosszú, mint azok.
A program a játék tábláját egy 8×8-as T% tömbben tárolja (ábra a köv. o.), melynek sorai a tábla jobbra felfelé haladó vonalainak, oszlopai pedig a tábla vízszintes sorainak felelnek meg. A tömb egy elemének értéke 0, ha a megfelelő helyen nincs korong, s+1 ill. –1, ha ott fekete, ill. fehér korong van. Ennek a +–1-es tárolásnak hatalmas előnye, hogy egy üres lyuk 6 szomszédjának a tartalmát összeadva az eredmény előjele megmutatja, hogy ki forgathat ott. A kiindulási helyzet a 4–14-es DATA sorokban van leírva. A megjelenítés karaktergrafikával történik, s emiatt a tábla képe egy kicsit torz, viszont a megjelenítés nagyon gyors. Mivel (főleg a karaktergrafika miatt) a program listája „tele van" grafikus karakterekkel, ezért a megfelelő sorok végére megjegyzésként odaírtuk, hogy az ott szereplő grafikus karakterek hogyan állíthatók elő. Ha egy jel több egymás utáni sorban is szerepel, akkor csak az első sorba írtuk ki a jelentését; a C< a commodore-billentyűt jelöli. Természetesen ezeket a jelmagyarázó REM részeket begépeléskor érdemes elhagyni.

A program játékstratégiája elég egyszerű. Először is a tábla egyes mélyedéseihez az ábrán látható alapértékeket rendeli. Ezek az alapértékek azt mutatják, hogy mennyire fontos az a mező, mennyire kell törekednünk az illető mező golyóval történő elfoglalására. Az állások kiértékeléséhez egy értékelő függvényt használ, mely egy játékállást úgy értékel ki, hogy összeadja az egyes mezők tartalmának és alap-értékének a szorzatát. (Emlékezzünk rá: –1-et akkor tartalmaz egy mező, ha az adott helyen fehér golyó van, +1-et pedig akkor, ha fekete; 0 akkor szerepel táblázatunkban, ha üres a mező.)
Ugyanezt végrehajthatjuk úgy is, hogy megszámoljuk, mennyivel több fekete golyó van a 2-es értékű lyukakban, mint fehér. (Ez a különbség lehet negatív is.) Az így kapott számot kettővel megszorozzuk, majd ugyanezt kiszámítjuk a 4-es, 8-as, ill. 33-as értékű lyukakra is, – itt, persze, értelemszerűen a 4-gyel, 8-cal, ill. 33-mal szorozzuk meg a különbséget, és ezután a négy beszorzott értéket összeadjuk.
Mivel a gép (ha játszik) a világos golyókat kezeli, és ezeket a T% tömbben –1 jelzi, így a lehetséges lépések közül a gép azt fogja lépni, amely a lehető legkisebb értékű helyzetet állítja elő. (Ez a végigpróbálgatás a 302–336 sorokban található, egy helyzet kiértékelését a 358–362 rutin végzi.)
Fontos észrevétel: mivel azt a lépést keressük, amely a pillanatnyi helyzet értékén minél többet

```
166 CHAR,35,23,B$+"-"
168 GETKEYB$:IF B$<"0"OR B$>"3"THEN168
169 IFB$="0"THENCHAR,35,23,"   ":GOTO158
170 CHAR,37,23,B$:F=VAL(B$)
172 FORI=1 TO F
174 GOSUB 228
176 GOSUB262:SOUND 1,770+20*I,3:NEXTI
178 :
180 FORI=1TO200:NEXTI
182 GOSUB290: GOSUB276
184 CHAR,0,22,BL$:CHAR,0,23,BL$
186 GOSUB292:REM KIERTEKELESHEZ
188 IF GY THEN CHAR,2,22,SZ$(GY+2)+" JATEKOS NYERT !":GOTO194
190 :
192 G=G*-1:GOTO136
194 VOL7:FORI=1TO4:SOUND1,560+50*I,10+2*I:NEXTI
196 CHAR,2,23,"AKARSZ UJRA JATSZANI? (I/N)"
198 GETKEYB$:IFB$="I"THEN206
200 IF B$<>"N" THEN198
202 CHAR,0,23,BL$
204 CHAR,8,23,"HA NEM HAT NEM!":END
206 RESTORE:FORI=0TO8:FORJ=0TO8:READA
208 T%(I,J)=A:NEXTJ,I
210 GOSUB106:GOTO134
212 :
214 REM *** LETEHETO? ***
216 N=0:IFX+Y<4ORX+Y>11THENRETURN
218 IF T%(X,Y)<>0THENRETURN
220 SZ=T%(X,Y+1)+T%(X,Y-1)+T%(X-1,Y)+T%(X-1,Y+1)+T%(X+1,Y)+T%(X+1,Y-1)
222 IF SZ>0AND G=1THEN N=1:RETURN
224 IF SZ<0AND G=-1 THEN N=1:RETURN
226 RETURN
228 Z%=T%(X,Y+1):REM ***FORGATAS***
230 T%(X,Y+1)=T%(X-KF,Y-(KF=1))
232 T%(X-KF,Y-(KF=1))=T%(X-KF,Y+(KF=-1))
234 T%(X-KF,Y+(KF=-1))=T%(X,Y-1)
236 T%(X,Y-1)=T%(X+KF,Y+(KF=1))
238 T%(X+KF,Y+(KF=1))=T%(X+KF,Y-(KF=-1))
240 T%(X+KF,Y-(KF=-1))=Z%
242 RETURN
260 REM *** UJ ALLAPOT RAJZOLAS ***
262 CHAR,X%(X,Y+1),2*(9-(Y+1)),G$(T%(X,Y+1)+2)
264 CHAR,X%(X-1,Y+1),2*(9-(Y+1)),G$(T%(X-1,Y+1)+2)
266 CHAR,X%(X-1,Y),2*(9-Y),G$(T%(X-1,Y)+2)
268 CHAR,X%(X,Y-1),2*(9-(Y-1)),G$(T%(X,Y-1)+2)
270 CHAR,X%(X+1,Y-1),2*(9-(Y-1)),G$(T%(X+1,Y-1)+2)
272 CHAR,X%(X+1,Y),2*(9-Y),G$(T%(X+1,Y)+2)
274 RETURN
276 CHAR,9,9,CHR$(129)+LEFT$(F$,17)+"       "+LEFT$(L$,18)
278 CHAR,9,11,LEFT$(L$,17)+"      "+LEFT$(F$,18)
280 CHAR,16,10,LEFT$(F$,7)+" "+LEFT$(L$,8)
282 CHAR,16,11,"————————————"+CHR$(144) :REM "—"=SHIFT C
284 RETURN
286 CHAR,X%(X,Y),2*(9-Y),CHR$(144)+KO$:RETURN
288 CHAR,X%(X,Y),2*(9-Y),CHR$(5)+KO$:RETURN
290 CHAR,X%(X,Y),2*(9-Y),CHR$(129)+K0$:RETURN
292 SZ=T%(3,4)+T%(3,5)+T%(4,4):REM ERT
294 IF SZ=3 THEN GY=1:GOTO300
296 IF SZ=-3THEN GY=-1:GOTO300
298 GY=0
300 RETURN
302 CS=-9999:KF=1:PR=36
304 RESTORE 28
306 CHAR,2,23,"▒AZ EN LEPESEM: " :REM "▒"=C< 2
308 IFLS<6THENPR=30
310 FORI=1TOPR:READX,Y:GOSUB216
312 IFN=0 THEN330
316 GOSUB358:PE=E
318 FORJ=1TO5:GOSUB228
320 GOSUB358:C=PE-E:IF C<=CS THEN 326
322 A=X:B=Y:F=J:CS=C
324 GOSUB292:IFGY=-1THENI=37
326 NEXTJ
328 GOSUB228
330 NEXTI
336 IF F>3 THEN F=6-F:KF=-1
338 :
340 X=A:Y=B:B$=CHR$(64+X)+"-"+STR$(Y)
342 CHAR,16,23,B$
344 IF KF=1 THENB$="JOBBRA"
346 IF KF=-1 THENB$="BALRA "
348 CHAR,21,23,"FORGATAS "+B$+STR$(F)
350 FORI=1TO300:NEXTI
352 GOSUB288
354 FORI=1TO1500:NEXTI
356 GOTO172
358 E=P%(X,Y+1)*T%(X,Y+1)+P%(X+1,Y)*T%(X+1,Y)+P%(X+1,Y-1)*T%(X+1,Y-1)
360 E=E+P%(X,Y-1)*T%(X,Y-1)+P%(X-1,Y)*T%(X-1,Y)+P%(X-1,Y+1)*T%(X-1,Y+1)
362 RETURN
```



csökkent, így csak a két helyzet értékének a különbségére vagyunk mindig kíváncsiak. Ehhez pedig elég az éppen próbált letevés körüli értékeket kiszámítani – hiszen a tábla többi mezőjének értéke úgysem változik. Tehát egy forgatás akkor jó, ha a körülötte álló fehér golyókat minél nagyobb, a feketéket pedig minél kisebb értékű lyukakba forgatja át. Pl. a 2. ábrán a C3 lyuk körüli részt látjuk, az alapértékeket a lyukak mellé írtuk. Ha most a forgatót C3-ra helyezzük, s egyet jobbra forgatunk, akkor a fehér golyók a 8, 0, 4 értékű lyukakból a 33, 8, 4 értékűekbe vándorolnak, míg a feketék a 33 és 8 értékűből a 0 és 4 értékűbe, így az állás értéke 33–8+8–0+4–4–(0–33+4–8)=70-nel javul (azaz a gépben 70-nel csökken).
A program kezelése igen egyszerű, s futás elején ki is íródik a képernyőre. A korong letevésének helyét az ábrán vastag betűkkel szedett koordináták segítségével adhatjuk be – ezek játék közben a képernyőn is láthatók. Az elforgatást az irány (B=balra, J=jobbra) és egy 1 és 3 közti szám segítségével adhatjuk be, mely szám azt mutatja, hogy hányszor hatvan fokkal akarjuk a korongot elforgatni. Mindenkinek jó játszadozást kívánunk!



TÁBLÁZATKEZELŐ PROGRAM ZX SPECTRUMRA

Sokáig gondolkoztunk, hogy közöljük-e következő programot. Hogy mégis a közlés mellett döntöttünk, oka az, hogy látszólag profibb, teljesen gépi kódban megírt adatbázis-kezelő programok léteznek Spectrumra is, de ez a program sok tanulsággal szolgálhat annak, aki egy kis fáradságot vesz és átbogarássza. Minthogy a program BASIC-ben van megírva, sebessége természetesen javítható valamilyen compiler használatával (a szerző a lefordított programot is bemutatta, és tényleg állíthatjuk, hogy sebessége nem hagy kívánnivalókat maga után).
A másik programrész nemcsak az 5 pixel széles karakterek kiírására alkalmas, hanem 4, 6 vagy 7 pixelre is (4 alatt a karakter nem olvasható, 8 pixel pedig az eredeti kiírórutin) a táblázat átírásával. Terjedelmi okokból döntöttünk úgy, hogy a leírásnál részletesebben ennek módját nem ismertetjük.

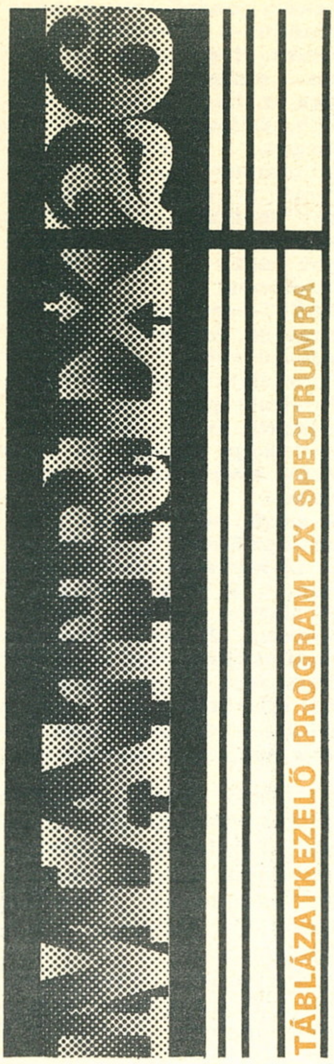
Az alábbi BASIC program 26x26 elemből álló táblázat kezelésére alkalmas, amelynek minden elemébe 26 karakternyi információ kerülhet. (szöveg, szám, képlet)
A képernyőn egyszerre 21 sor és 6 oszlop fér el, amelynek minden boxában 8 karakter látható.
A képletekben – hasonlóan a VU-calc programhoz – hivatkozhatunk egyes boxokra, de ez a program a matematikai műveletek szélesebb skálájával rendelkezik.

VEZÉRLÉS: az alábbi billentyűk megnyomásával.
ENTER: szám, képlet bevitele az aktuális boxba. Befejezés: ENTER
SPACE: szöveg bevitele az aktuális boxba. Befejezés: ENTER
5, 6, 7, 8: a kurzor mozgatása a boxokon, vagyis az aktuális box kijelölése.
0: az aktuális box törlése.
1: editálás, vagyis az aktuális box tartalmának javítása, szerkesztése. A szerkesztő a CAPS–5,6,7,8 billentyűkkel működtethető úgy, hogy CAPS 5,8 mozgatja a kurzort, CAPS–7 hézagot készít, CAPS–6 töröl. Befejezés: ENTER.
A: az egész táblázat törlése.
B: Kék alapszínű tartomány létrehozása az aktuális box felett és tőle balra, amelyre nem vihető a kur-

1. LISTA

```
   0 REM [illegible]
  20 CLS : GO SUB 5000
  30 PRINT #9; INK 2;AT 20,16;"E
NTER-SPACE (5,6,7,8) Command 01A
BCDFGHIJRSZ "
  35 IF PEEK 23200=70 THEN GO TO
 20
  40 LET l=l+(k$="6")-(k$="7")
  50 LET c=c+5*((k$="8")-(k$="5"
))
  53 IF c>27 AND c>=c1 THEN LET
c=27: LET u=u+1: GO SUB 5050
  54 IF c<2+(tv*5) AND c<=c1 THE
N LET c=2+(tv*5): LET u=u-1: GO
SUB 5050
  60 IF l<1+th AND l<=l1 THEN LE
T l=1+th: LET v=v-1: GO SUB 5010
  62 IF l>21 AND l>=l1 THEN LET
l=21: LET v=v+1: GO SUB 5010
  90 PRINT OVER 1; PAPER 0;AT l1
,c1;"     "
  92 LET l1=l: LET c1=c
  93 LET co=INT ((c-2)/5)+u
  94 LET lo=l+v-1
  95 PRINT INK 0; OVER 1; PAPER
4;AT l,c;"     "
  96 PRINT #9; PAPER 0;AT 22,16;
"
                 "
  98 PRINT #9; INK 2;AT 22,16;d$
(lo,co)
 100 PRINT #9; INK 2;AT 22,240;C
HR$ (64+lo);co;" "
 110 BEEP .01,66: PAUSE 0: LET k
$=INKEY$: BEEP .01,50
 120 IF k$="0" OR k$=CHR$ 32 OR
k$=CHR$ 13 THEN GO TO 7000
 130 IF k$>CHR$ 96 AND k$<CHR$ 1
23 THEN GO TO 7300
 140 IF k$="1" THEN GO TO 2500
 150 GO TO 40
 301 PRINT #9;AT 22,208;"calc  "
 310 FOR i=1 TO cl: FOR j=1 TO l
l: LET co=i: LET lo=j
 311 LET a$=d$(j,i,2 TO )
 312 IF d$(j,i,1)<>"=" THEN GO T
O 320
 315 GO SUB 3900: GO SUB 4050
 320 NEXT j: NEXT i
 350 RETURN
 410 INPUT "Beszuras sor v.oszlo
p?(s/o)";g$
 415 IF g$="" THEN RETURN
 416 PRINT #9;AT 22,208;"insert"
 420 IF g$="s" THEN GO TO 500
 425 IF cl=26 THEN RETURN
 430 FOR n=1 TO ll: FOR j=cl TO
1 STEP -1: LET i$=d$(n,j)
 450 LET k=2: LET ku=1: GO SUB 9
00
 460 IF j<co THEN LET d$(n,j)=i$
: GO TO 480
 465 IF i$(1)<>"=" THEN LET s$(n
,j+1)=i$(2 TO )
 470 LET d$(n,j+1)=i$: IF j=co T
HEN LET d$(n,j)="": LET s$(n,j)=
"": LET xx=0: LET lo=n: GO SUB 4
000
 485 NEXT j: NEXT n: LET cl=cl+1
: GO SUB 300: GO SUB 300: GO SUB
 5010: RETURN
 500 IF ll=25 THEN RETURN
 520 FOR n=1 TO cl: FOR j=ll TO
1 STEP -1: LET i$=d$(j,n)
 540 LET k=2: LET ku=1: GO SUB 9
50
 550 IF j<lo THEN LET d$(j,n)=i$
: GO TO 570
 555 IF i$(1)<>"=" THEN LET s$(j
+1,n)=i$(2 TO )
 560 LET d$(j+1,n)=i$: IF j=lo T
HEN LET d$(j,n)="": LET s$(j,n)=
"": LET xx=0: LET co=n: GO SUB 4
000
 570 NEXT j: NEXT n: LET ll=ll+(
ll<25): GO SUB 300: GO SUB 300:
GO SUB 5010: RETURN
 705 INPUT "Torles sor v.oszlop?
(s/o)";g$: IF g$="" THEN RETURN
 710 PRINT #9;AT 22,208;"delete"
 720 IF g$="s" THEN GO TO 800
 730 FOR n=1 TO ll: FOR j=co TO
cl: LET i$=d$(n,j+(j<26)): IF i$
(1)<>"=" THEN LET s$(n,j)=i$(2 T
O ): GO TO 750
 740 LET k=2: LET ku=-1: GO SUB
900
 750 LET d$(n,j)=i$: NEXT j
 760 LET xx=0: LET lo=n: GO SUB
4000: NEXT n: LET cl=cl-1
 770 FOR n=1 TO ll: FOR j=1 TO c
o-1: LET i$=d$(n,j)
 775 LET k=2: LET ku=-1: GO SUB
900
 780 LET d$(n,j)=i$: NEXT j
 790 NEXT n: GO SUB 300: GO SUB
300: GO SUB 5010: RETURN
 800 FOR n=1 TO cl: FOR j=lo TO
ll: LET i$=d$(j+(lo<25),n): IF i
$(1)<>"=" THEN LET s$(j,n)=i$(2
TO ): GO TO 870
 820 LET k=2: LET ku=-1: GO SUB
950
 830 LET d$(j,n)=i$: NEXT j
 840 LET xx=0: LET co=n: GO SUB
4000: NEXT n: LET ll=ll-1
 850 FOR n=1 TO cl: FOR j=1 TO l
o-1: LET i$=d$(j,n)
 860 LET k=2: LET ku=-1: GO SUB
950
 870 LET d$(j,n)=i$: NEXT j
 890 NEXT n: GO SUB 300: GO SUB
300: GO SUB 5010: RETURN
 900 IF i$(1)<>"=" THEN RETURN
 903 IF i$(9)=":" THEN GO TO 930
 905 IF i$(k)>CHR$ 64 AND i$(k)<
CHR$ 91 AND i$(k+1)="(" THEN GO
TO 910
 906 GO TO 945
 910 IF VAL i$(k+2 TO k+3)>=co T
HEN LET i$(k+2 TO k+3)=STR$ (VAL
 (i$(k+2 TO k+3))+ku)
 920 IF i$(k+3)=" " THEN LET i$(
k+3)=i$(k+2): LET i$(k+2)="0"
 925 GO TO 945
 930 IF VAL i$(7 TO 8)>=co THEN
LET i$(7 TO 8)=STR$ (VAL (i$(7 T
O 8))+ku)
 933 IF VAL i$(11 TO 12)>=co THE
N LET i$(11 TO 12)=STR$ (VAL (i$
```

```
(11 TO 12))+ku)
 940 IF i$(8)=" " THEN LET i$(8)=i$(7): LET i$(7)="0"
 942 IF i$(12)=" " THEN LET i$(12)=i$(11): LET i$(11)="0"
 945 IF i$(k)=" " OR k=23 OR i$(9)=":" THEN RETURN
 947 LET k=k+1: GO TO 900
 950 IF i$(1)<>"=" THEN RETURN
 955 IF i$(9)=":" THEN GO TO 970
 960 IF i$(k)>=CHR$ (64+lo) AND i$(k)<"Z" AND i$(k+1)="(" THEN LET i$(k)=CHR$ (CODE i$(k)+ku)
 965>GO TO 990
 970 IF i$(6)>=CHR$ (64+lo) THEN LET i$(6)=CHR$ (CODE i$(6)+ku)
 980 IF i$(10)>=CHR$ (64+lo) THEN LET i$(10)=CHR$ (CODE i$(10)+ku)
 990 IF i$(k)=" " OR k=23 OR i$(9)=":" THEN RETURN
 995 LET k=k+1: GO TO 950
1010 LET f1=CODE a$(5)-64
1020 LET v1=VAL a$(6 TO 7)
1030 LET f2=CODE a$(9)-64
1040 LET v2=VAL a$(10 TO 11)
1042 IF ll<f2 THEN LET ll=f2
1043 IF cl<v2 THEN LET cl=v2
1045 IF s$(f1,v1,1)=" " THEN LET xx=0: GO TO 1055
1050 LET xx=VAL s$(f1,v1)
1070 IF a$(1 TO 3)="ATL" THEN GO TO 2000
1090 IF a$(1 TO 3)="MIN" THEN GO TO 1500
1100 FOR F=f1 [illegible]
1110 IF s$(F,D,1)=" " THEN GO TO 1200
1120 IF VAL s$(F,D)>xx THEN LET xx=VAL s$(F,D)
1200 NEXT D: NEXT F: GO TO 4000+lo
1500 FOR F=f1 TO f2: FOR D=v1 TO v2
1550 IF s$(F,D,1)=" " THEN LET xx=0: GO TO 1610
1600 IF VAL s$(F,D)<xx THEN LET xx=VAL s$(F,D)
1700 NEXT D: NEXT F: GO TO 4000+lo
2000 LET xx=0: LET sza=0
2050 FOR F=f1 TO f2: FOR D=v1 TO v2
2070 IF s$(F,D,1)=" " THEN LET sza=sza+1: GO TO 2200
2100 LET sza=sza+1: LET xx=xx+VAL s$(F,D)
2200 NEXT D: NEXT F: LET xx=xx/sza: GO TO 4000+lo
2500 PRINT #9; INK 0; PAPER 3;AT 22,16;d$(lo,co)
2510 LET e$=d$(lo,co): LET n=2: LET hj=2
2520 PRINT #9; INK 0; PAPER 3;AT 22,11+5*hj;"_"
2540 PAUSE 0: LET k$=INKEY$
2542 PRINT #9; INK 0; PAPER 3;AT 22,11+5*hj;e$(n)
2543 IF k$=CHR$ 14 THEN PAUSE 0: LET k$=INKEY$: GO SUB 2800
2545 IF k$>CHR$ 31 THEN LET e$(n)=k$: PRINT #9; INK 0; PAPER 3;AT 22,11+5*hj;k$
2550 IF k$=CHR$ 13 THEN GO TO 2600
2555 LET k=(k$>CHR$ 31 AND n<26)+(k$=CHR$ 9)-(k$=CHR$ 8)
2557 IF k$=CHR$ 10 OR k$=CHR$ 11 THEN LET k=(k$=CHR$ 11)*(n<27)-(k$=CHR$ 10)*(n>2): LET e$(n+k TO )=e$(n TO ): LET e$(n)=CHR$ (32*(k=1)+(CODE e$(n))-CODE e$(n)*(k=1)): PRINT #9; INK 0; PAPER 3;AT 22,16;e$
2562 IF k=1 AND e$(n)>CHR$ 164 THEN LET hj=hj+k*3-(e$(n)=CHR$ 167)-(e$(n)=CHR$ 184)
2565 LET n=n+k: LET hj=hj+k
2582 IF n=28 THEN LET n=n-1: LET hj=hj-1
2590 IF k=-1 AND e$(n)>CHR$ 164 THEN LET hj=hj+k*3+(e$(n)=CHR$ 167)+(e$(n)=CHR$ 184)
2592 IF n<3 THEN LET n=2: LET hj=2
2595 GO TO 2520
2600 BEEP .02,55
2605 LET d$(lo,co)=e$: LET a$=e$(2 TO ): LET s$(lo,co)=a$
2610 IF e$(1)="=" THEN GO SUB 3900: GO SUB 4050
2620 GO SUB 4100: GO TO 30
2800 IF k$="q" THEN LET k$=CHR$ 178
2805 IF k$="w" THEN LET k$=CHR$ 179
2810 IF k$="e" THEN LET k$=CHR$ 180
2815 IF k$="r" THEN LET k$=CHR$ 186
2820 IF k$="f" THEN LET k$=CHR$ 188
2825 IF k$="g" THEN LET k$=CHR$ 189
2830 IF k$="h" THEN LET k$=CHR$ 187
2835 IF k$="x" THEN LET k$=CHR$ 185
2840 IF k$="z" THEN LET k$=CHR$ 184
2845 IF k$="m" THEN LET k$=CHR$ 66
2850 IF k$="j" THEN LET k$=CHR$ 175
2855 IF k$="k" THEN LET k$=CHR$ 177
2860 IF k$="l" THEN LET k$=CHR$ 165
2865 IF k$="<=" THEN LET k$=CHR$ 181
2870 IF k$="<>" THEN LET k$=CHR$ 182
2875 IF k$=">=" THEN LET k$=CHR$ 183
2890 RETURN
3000 POKE 23658,8: INPUT "Ugras A1-Z26 boxba ? "; LINE g$: POKE 23658,0
3005 IF g$="" THEN RETURN
3006 PRINT #9;AT 22,200;"goto ";g$
3010 IF g$(1)>CHR$ 90 OR g$(1)<CHR$ 65 OR VAL g$(2 TO )>34 THEN
```



zor, és mindig látható marad. Megszüntetése B újbóli megnyomásával lehetséges.

C: újraszámítja a teljes táblázatot.

D: sor vagy oszlop kitörlése a táblázatból.

F: a számok kijelzési formátumának beállítása.

G: ugrás az A1–Z26 box bármelyikébe.

H: help

I: sor vagy oszlop beszúrása

J: táblázat betöltése kazettáról.

R: az aktuális box tartalmának ismétlése vízszintes vagy függőleges irányban abszolút vagy relatív módon.

S: a táblázat kimentése kazettára.

Z: a képernyő kinyomtatása ZX printerre.

SZINTAXIS:

Box megadása képletekben: X(nn) Pl: A(01)

Függvények bevitele a funkcióbillentyűkkel lehetséges (CAPS–SYM SH).

```
GO TO 3500
3010>IF g$(1)>CHR$ 90 OR g$(1)<CHR$ 65 OR VAL g$(2 TO )>34 THEN GO TO 3500
3050 LET sor=CODE g$(1)-64: LET oszl=VAL g$(2 TO )
3060 FOR i=u+tv TO y-(cl<y)*(y-cl-1)
3065 FOR f=1+th TO (ll<21)*(ll+1)+(ll>=21)*21
3070 PRINT AT f,(i-u)*5+2;"     "
3080 NEXT f: NEXT i
3090 LET l=l+sor-lo
3100 IF sor>x THEN LET v=v+sor-x: LET l=21
3110 IF sor<v+th THEN LET v=sor-tv: LET l=1
3115 IF l<=th THEN LET l=1+th
3130 LET c=c+(oszl-co)*5
3140 IF oszl>u+5 THEN LET u=oszl-5: LET c=27
3150 IF oszl<u+tv THEN LET u=oszl-tv: LET c=2
3160 IF (c-2)/5<tv THEN LET c=2+tv*5
3200 GO SUB 5010
3500 GO TO 30
3910 IF a$(1 TO 3)="MAX" OR a$(1 TO 3)="MIN" OR a$(1 TO 3)="ATL" THEN GO TO 1000
3999 LET xx=VAL a$
4000 GO TO 4000+lo
4001 LET A(co)=xx: GO TO 4030
4002 LET B(co)=xx: GO TO 4030
4003 LET C(co)=xx: GO TO 4030
4004 LET D(co)=xx: GO TO 4030
4005 LET E(co)=xx: GO TO 4030
4006 LET F(co)=xx: GO TO 4030
4007 LET G(co)=xx: GO TO 4030
4008 LET H(co)=xx: GO TO 4030
4009 LET I(co)=xx: GO TO 4030
4010 LET J(co)=xx: GO TO 4030
4011 LET K(co)=xx: GO TO 4030
4012 LET L(co)=xx: GO TO 4030
4013>LET M(co)=xx: GO TO 4030
4014 LET N(co)=xx: GO TO 4030
4015 LET O(co)=xx: GO TO 4030
4016 LET P(co)=xx: GO TO 4030
4017 LET Q(co)=xx: GO TO 4030
4018 LET R(co)=xx: GO TO 4030
4019 LET S(co)=xx: GO TO 4030
4020 LET T(co)=xx: GO TO 4030
4021 LET U(co)=xx: GO TO 4030
4022 LET V(co)=xx: GO TO 4030
4023 LET W(co)=xx: GO TO 4030
4024 LET X(co)=xx: GO TO 4030
4025 LET Y(co)=xx: GO TO 4030
4026 LET Z(co)=xx: GO TO 4030
4031 RETURN
4050 LET s$(lo,co)=STR$ (INT (a*xx+.5)/a)
4052 IF ABS xx<.1 THEN LET s$(lo,co)=STR$ xx
4055 IF ABS xx>=10↑8 OR ABS xx<=10↑-6 AND xx<>0 THEN LET x$=STR$ xx: LET y=LEN x$: LET s$(lo,co)=x$( TO 4): IF y>5 THEN LET s$(lo,co,8 TO )=x$(y-3 TO y)
4060 RETURN
4100 PRINT AT l,c;"     ": PRINT #9;AT l,c*8;s$(lo,co)
4110 RETURN
5000 IF tv<>0 AND th<>0 THEN FOR n=0 TO th: FOR m=0 TO tv-1: PRINT #9; PAPER 1;AT n,m*40+16;s$(n,m+1): NEXT m: NEXT n
5010 IF v>=6 THEN LET v=6
5020 IF v<=1 THEN LET v=1
5030 FOR g=th TO 20: PRINT #9; PAPER 2;AT g+1,0;"  ";CHR$ (g+v+64): NEXT g
5050 IF u>=21 THEN LET u=21
5060 IF u<=1 THEN LET u=1
5070 PRINT PAPER 2;AT 0,0;"                                ": FOR j=tv TO 5: PRINT #9; PAPER 2;AT 0,j*40+30;j+u;" ": NEXT j
5075 IF tv<>0 THEN FOR j=0 TO tv-1: PRINT #9; PAPER 2;AT 0,40*j+30;j+1;" ": NEXT j: FOR n=1+th TO 21: FOR m=0 TO tv-1: PRINT PAPER 1;AT n,m*5+2;"     ": PRINT #9; PAPER 1;AT n,m*40+16;s$(n+v-1,m+1): NEXT m: NEXT n
5080 IF th<>0 THEN FOR g=1 TO th: PRINT #9; PAPER 2;AT g,0;"  ";CHR$ (g+64);" ": NEXT g: FOR n=1 TO th: FOR m=tv TO 5: PRINT PAPER 1;AT n,m*5+2;"     ": PRINT #9; PAPER 1;AT n,m*40+16;s$(n,m+u): NEXT m: NEXT n
5105 LET x=v+20: LET y=u+5
5110 FOR i=u+tv TO y-(cl<y)*(y-cl-1)
5120 FOR f=1+th TO (ll<21)*(ll+1)+(ll>=21)*21
5125 PRINT AT f,(i-u)*5+2;"     "
5130 PRINT #9;AT f,(i-u)*40+16;s$(f+v-1,i)
5140 NEXT f: NEXT i
5150 RETURN
6000 LET z$="": LET i$=d$(lo,co)
6010 INPUT "Repl. vizsz.v.fugg.? (v/f) ";g$
6011 IF g$="" THEN RETURN
6012 IF i$(1)="=" THEN INPUT "Abszolut v. relativ ?(a/r) ";z$
6015 INPUT "Hany boxban ismetlodjon? ";n: PRINT #9;AT 22,208;"repl"
6020 IF g$="f" THEN GO TO 6100
6030 IF co+n>26 THEN LET n=26-co
6035 FOR j=co+1 TO co+n
6040 IF i$(1)<>"=" OR z$="a" THEN GO TO 6075
6050 LET k=2: LET ku=1: GO SUB 900
6075 LET s$(lo,j)=i$(2 TO )
6080 LET d$(lo,j)=i$: NEXT j: IF j>cl THEN LET cl=j-1
6090 GO SUB 5100: RETURN
6100 IF lo+n>26 THEN LET n=26-lo
6110 FOR j=lo+1 TO lo+n
6120 IF i$(1)<>"=" OR z$="a" THEN GO TO 6150
6130 LET k=2: LET ku=1: GO SUB 950
6150 LET s$(j,co)=i$(2 TO )
6160 LET d$(j,co)=i$: NEXT j: IF j>ll THEN LET ll=j-1
6170 GO SUB 5100: RETURN
7001 IF k$="0" THEN GO TO 8500
7005 IF k$=CHR$ 13 THEN LET k$="
```

```
="
7010 IF k$=CHR$ 32 THEN LET k$=":"
7020 DIM a$(25)
7050 IF k$="=" THEN POKE 23658,8: INPUT PAPER 2;"Value "; LINE a$
7055 IF k$="""" THEN INPUT PAPER 2;"Label "; LINE a$
7060 POKE 23658,0
7065 IF a$(1)=" " THEN GO TO 30
7074 IF lo>ll THEN LET ll=lo
7075 IF co>cl THEN LET cl=co
7090 LET d$(lo,co)=k$+a$: LET s$(lo,co)=a$
7100 IF k$="=" THEN GO SUB 3900: GO SUB 4050
7150 GO SUB 4100: GO TO 30
7310 POKE 23658,0
7320 IF k$="g" THEN GO SUB 3000
7329 IF k$<>"b" THEN GO TO 7340
7330 IF th=0 AND tv=0 THEN LET tv=co-u: LET th=lo-v: LET thh=th: LET tvv=tv: LET pap=1: GO TO 7336
7333 LET pap=0: LET tv=0: LET th=0
7336 FOR n=1 TO thh: PRINT OVER 1; PAPER pap;AT n,2;"  ": NEXT n: FOR n=1 TO 21: FOR m=0 TO tvv-1: PRINT OVER 1; PAPER pap;AT n,m*5+2;"  ": NEXT m: NEXT n
7340 IF k$="z" THEN COPY
7350 IF k$="r" THEN GO SUB 6000
7355 IF k$="c" THEN GO SUB 300: GO SUB 5100
7360 IF k$="i" THEN GO SUB 400
7365 IF k$="d" THEN GO SUB 700
7370 IF k$="f" THEN INPUT "Tizedesek szama? ",p: LET a=10↑p: GO SUB 300: GO SUB 5100
7376 IF k$="a" THEN INPUT "Tablazat torles biztos?(i/n)";z$: IF z$="i" THEN PRINT #9;AT 22,208;"TORLES": GO SUB 9110: GO TO 10
7377 IF k$="h" THEN PRINT #9;AT 22,0;"0=blank 1=edit A=new B=border C=calc D=delet F=form";AT 23,0;"G=goto H=help I=insert J=load R=repl. S=save Z=copy": PAUSE 800: FOR f=23232 TO 23295: POKE f,0: NEXT f
7385 IF k$="j" THEN INPUT "Tablazat neve ?";g$: IF g$<>"" THEN GO SUB 9100: LOAD g$ DATA d$(): GO SUB 8000: GO TO 10
7386 IF k$="s" THEN INPUT "Tablazat neve:";g$: IF g$<>"" THEN SAVE g$ DATA d$()
7400 GO TO 30
8000 CLS : PRINT "        TABLAZAT FELTOLTESE !"
8005 FOR s=1 TO 26
8010 FOR o=1 TO 26
8040 IF d$(s,o,1)<>" " AND cl<o THEN LET cl=o
8045 IF d$(s,o,1)<>" " AND ll<s THEN LET ll=s
8050 LET a$=d$(s,o,2 TO ): LET s$(s,o)=a$
8090 NEXT o: NEXT s: RETURN
8500 LET xx=0: GO SUB 4000: LET d$(lo,co)="": LET s$(lo,co)="": GO TO 7110
9100 PAPER 0: INK 5: BORDER 0: BRIGHT 1: CLS
9110 LET a=100: LET k$="": LET th=0: LET tv=0: LET v=1: LET u=1: LET ll=1: LET cl=1: LET l1=1: LET c1=2: LET l=1: LET c=2
9150 LET co=26: LET lo=26: LET w=26
9200 DIM s$(lo,co,8): DIM A(co): DIM B(co): DIM C(co): DIM D(co): DIM E(co): DIM F(co): DIM G(co): DIM H(co): DIM I(co): DIM J(co): DIM K(co): DIM L(co): DIM M(co): DIM N(co): DIM O(co): DIM P(co): DIM Q(co): DIM R(co): DIM S(co): DIM T(co): DIM U(co): DIM V(co): DIM W(co): DIM X(co): DIM Y(co): DIM Z(co): DIM d$(lo,co,w): RETURN
9400 LOAD ""CODE : RANDOMIZE USR 63744
9500 GO SUB 9100: GO TO 10
9999 SAVE "matrix" LINE 9400
```

2. LISTA

```
10 CLS : CLEAR 63743: RESTORE 1000: LET sz=0
50 FOR n=63744 TO 65279: READ x: POKE n,x: LET sz=sz+x: NEXT n
100 IF sz<>141242 THEN PRINT AT 10,0;"HIBAS ADATOK!": STOP
200 RANDOMIZE USR 63744: PRINT #9; AT 8,20;
250 FOR f=32 TO 127: PRINT #9;CHR$ f;" ";: NEXT f: STOP
1000 DATA 1,5,0,42,83,92,43,205,
85,22,35,235,213,33,34,249,1,5,0
,237,176,225,237,91,79,92,175,23
7,82,35,34,40,92,201,39,249,196,
21,88,217,213,217,205,49,249,217
,209,217,201,95,58,170,250,87,58
,41,251,178,32,10,123,254,165,21
[illegible],250,254,144,56,2,214,79,25
[illegible],56,2,62,35,95,33,180,250,5
[illegible],203,127,194,227,250,56,
[illegible],203,79,58,170,250,194,42,2
51,254,1,40,51,48,29,193,254,13,
[illegible],138,250,254,6,32,9,126,254,1
[illegible],210,[illegible],250,201,254,[illegible]
4,[illegible],135,230,248,214,6,5
4,0,35,119,62,1,50,170,250,201,
1010 DATA 175,50,170,250,123,190
,115,218,140,250,201,254,22,32,6
,62,2,50,170,250,201,254,165,210
```

vagyis a SPECTRUM szintaxisa szerint.

Használható függvények: sin, cos, tan, int, rnd, abs, val, sqr, len, ln, exp, arcsin, arccos, arctan, mint konstans a PI. **Matematikai műveletek:** +, –, /, ∗ és a hatványozás.

Ezen kívül maximum (MAX), minimum (MIN), átlag (ATL) függvény is használható, de olyan megkötéssel, hogy egy boxban csak egyedül állhat. Pl: ATL(A01:D04) ami az A1-es box és a D4-es box közé eső téglalap alakú tartomány értékeinek átlagát adja.

Szintaktikai hiba esetén történő programmegáláskor GOTO 1-gyel a program újra indítható.

A táblázatot egy gépi kódú program jeleníti meg a képernyőn, amely alkalmas 5 pixel szélességű karakterek kiírására a 0–23. sor 0–250. képpontjának bármelyikében, a 32–127 kódú karakterek tartományában.

A rutin a 63744(F900H) címen kezdődik, 696 byte hosszú és nem áthelyezhető. A karakterek kódtáblázata a 64512 címen a SPACE karakterrel indul és 65279-nél végződik.

Minden karaktert 8 byte ábrázol úgy, hogy az 1. byte a karakter szélességét jelenti, tehát az alakot 7 byte írja le a legfelső sor nélkül, ami zérus.

A program egyébként egy csatornát nyit meg a 9. stream-en, így egy – PRINT #9; – utasításban használható. Pl. PRINT #9; INK 0; PAPER 7; AT 53,125; "23. sor–125. pixel"

A betöltő program RUN-nal indítva elhelyezi a rutint és a karaktertáblát, majd kiírja a képre a 96 karaktert. Hibás beírás esetén figyelmeztet a hibára.

Felvétel kazettára: SAVE "51 kar" CODE 63744,1536

A főprogram eltárolását kazettára a GOTO 9999-el végezhetjük el.

Látható, hogy betöltés után 9400-on automatikusan indul, ahol várja a gépi rutin betöltését, majd aktivizálja az említett csatornát, feltölti a változókat és a program elején indul.

Nargang Vilmos
1045 Bp., Izzó u. 6/B

```
,171,250,214,32,218,221,250,205,
199,249,213,6,8,197,205,214,249,
24,64,56,0,111,41,41,41,17,0,252
,25,94,229,221,225,201,58,145,92
,6,255,31,56,1,4,31,31,246,120,1
4,128,61,203,41,61,32,251,121,203
0,255,217,87,30,255,58,166,250,2
30,7,40,8,55,203,26,203,27,61,32
.
1020 DATA 248,217,241,216,14,0,2
01,42,168,250,62,175,148,230,248
,95,203,63,203,63,203,63,230,24,
246,64,103,123,230,56,203,39,203
,39,203,61,203,61,203,61,181,111
,62,0,24,3,221,126,0,205,64,250,
35,241,61,245,32,244,24,57,62,0,
24,241,169,87,30,0,56,168,250,23
0,7,40,7,203,58,203,27,61,32,249
,217,122,217,166,119,122,160,32,
4,178,182,54,0,174,119,35,217,12
3,
1030 DATA 217,166,119,123,160,32
,4,179,182,54,0,174,119,43,221,3
5,201,193,37,229,0,205,219,11,22
5,205,219,11,209,33,168,250,123,
254,254,255,119,216,175,119,35,1
26,254,8,56,4,214,8,119,201,62,7
,119,42,81,92,229,6,2,0,0,0,225,
34,81,92,201,0,195,144,0,214,165
,216,17,149,0,245,205,65,12,56,7
,62,32,213,205,39,249,209,26,230
,127,213,205,39,249,209,26,19,
1040 DATA 135,48,243,209,254,72,
40,3,254,130,216,122,254,3,216,6
2,32,205,39,249,201,198,144,50,4
1,251,201,230,7,79,214,4,48,31,6
,0,33,33,251,9,70,5,40,4,123,23,
16,253,17,4,0,25,70,33,143,92,17
4,150,174,119,175,50,41,251,201,
60,79,123,254,2,46,244,33,145,92
,121,254,1,123,6,1,32,229,7,7,6,
4,24,223,10,4,8,7,7,56,128,64,0,
43,254,
1050 DATA 1,123,40,115,48,107,25
4,222,32,6,62,2,50,170,250,201,25
4,13,32,3,175,24,96,254,23,32,4,
62,1,24,237,254,6,32,7,126,230,1
28,238,128,24,77,214,32,218,221,
250,205,199,249,213,205,173,251,
33,0,91,58,168,250,203,63,203,63
,203,63,111,209,213,6,8,197,205,
214,249,62,0,24,3,221,126,0,205,
64,250,62,32,133,111,241,61,245,
32,241,193,33,167,250,205,173,
1060 DATA 251,209,123,134,119,25
4,248,216,175,119,195,205,14,62,
1,50,170,250,201,190,119,208,175
,50,170,250,195,205,14,237,91,16
7,250,122,83,95,237,83,167,250,2
01,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,
1070 DATA 0,0,0,0,0,0,0,0,0,0,
0,0,5,0,0,0,0,0,0,0,5,64,64,64,6
4,0,64,0,5,80,80,0,0,0,0,0,5,80,
248,80,80,248,80,0,5,16,120,80,1
20,40,120,32,5,72,72,16,32,72,72
,0,5,64,160,64,168,144,104,0,5,3
2,64,0,0,0,0,5,32,64,64,64,64,
32,0,5,64,32,32,32,32,64,0,5,0,1
60,64,224,64,160,0,
1080 DATA 5,0,32,32,112,32,32,
0,5,0,0,0,0,0,32,64,5,0,0,0,112,
0,0,0,5,0,0,0,0,0,32,0,5,16,16,3
2,32,64,64,0,5,96,144,176,208,14
4,96,0,5,32,96,32,32,32,112,0,5,
96,144,16,32,64,240,0,5,96,144,3
2,16,144,96,0,5,32,96,160,240,32
,32,0,5,112,64,96,16,144,96,0,5,
96,128,224,144,144,96,0,5,240,16
,32,
1090 DATA 64,64,64,0,5,96,144,
96,144,144,96,0,5,96,144,144,112
,16,96,0,5,0,0,32,0,0,32,0,5,0,0
,32,0,0,32,64,5,0,16,32,64,32,16
,0,5,0,0,112,0,112,0,0,5,0,64,32
,16,32,64,0,5,96,144,32,64,0,64,
0,5,96,144,176,176,128,112,0,5,9
6,144,144,240,144,144,0,5,224,14
4,224,144,144,224,0,5,96,144,128
,128,144,96,0,
1100 DATA 5,192,160,144,144,16
0,192,0,5,240,128,224,128,128,24
0,0,5,240,128,224,128,128,128,0,
5,96,144,128,176,144,96,0,5,144,
144,240,144,144,144,0,5,112,32,3
[illegible],112,0,5,48,16,16,16,144,[illegible]
[illegible]
,144,144,144,144,0,5,144,208,208
,176,176,144,0,5,96,144,144,144,
144,96,0,5,224,144,144,
1110 DATA 224,128,128,0,5,96,1
44,144,144,160,80,0,5,224,144,14
4,224,160,144,0,5,96,144,64,32,1
44,96,0,5,112,32,32,32,32,0,5
,144,144,144,144,144,96,0,5,144,
144,144,144,80,32,0,5,144,144,14
4,144,240,144,0,5,144,144,96,96,
144,144,0,5,144,144,96,32,32,32,
0,5,240,16,32,64,128,240,0,5,96,
64,64,64,64,96,0,5,64,64,32,32,1
6,16,0,
1120 DATA 5,96,32,32,32,32,96,
0,5,0,32,112,32,32,32,0,5,0,0,0,
0,0,0,240,5,48,80,192,96,128,240
,0,5,0,96,16,112,144,112,0,5,128
,128,224,144,144,224,0,5,0,96,14
4,128,144,96,0,5,16,16,112,144,1
44,112,0,5,0,96,144,224,128,112,
0,5,96,128,192,128,128,0,5,0
,112,144,144,112,16,224,0,5,128,22
4,144,144,144,144,0,5,32,0,96,
1130 DATA 32,32,112,0,5,32,0,9
6,32,32,32,64,5,128,144,160,192,
160,144,0,5,96,32,32,32,112,0
,5,0,208,176,144,144,144,0,5,0,2
24,144,144,144,144,0,5,0,96,144,
144,144,96,0,5,0,224,144,144,224
,128,128,5,0,112,144,144,112,16,
16,5,0,160,208,128,128,128,0,5,0
,96,128,96,16,224,0,5,64,64,224,
64,80,32,0,5,0,144,144,144,144,1
12,0,
1140 DATA 5,0,144,144,144,80,3
2,0,5,0,144,144,144,240,144,0,5,
0,144,96,96,144,144,0,5,0,144,14
4,144,112,16,224,5,0,240,16,96,1
28,240,0,5,16,32,32,224,32,16,0,
5,32,32,32,32,32,32,5,64,32,3
2,48,32,64,0,5,0,16,208,176,128,
2,5,112,136,184,200,184,136,11
2,
```

öttlet

A GEOS-RÓL

AZOKNAK, AKIK A GEOST MÉG NEM ISMERIK, ÉS AZOKNAK, AKIK A GEOST MÁR JÓL ISMERIK

1986 óta egy rendszerprogram hódította meg a C 64 kedvelőit, a GEOS.
Hogy mi is az a GEOS? Mint a legtöbb programnév, ez is betűszó: Grafics Enviropment Operating System, amelyet grafikus parancstáblával rendelkező grafikus operációs rendszernek fordíthatunk. A grafikus parancstábla alatt azt értjük, hogy nem kell parancsokat begépelni, hanem valamilyen ábra, felület kiválasztásával tudunk parancsot kiadni. Ezt vezérelhetjük joystickkal, egérrel és egyéb más bemeneti eszközökkel.

MIÉRT JÓ A GEOS?

1. Az összes lemezműveletet egyszerűen megvalósíthatjuk, ez nagyon emberségessé teszi a gép kezelését. Jó példa erre a file-ok törlése:
C 64 módban: OPEN15,8,15,"0:program név": CLOSE15 parancssort kell kiadnunk.
GEOS-ban: a képernyő alján található egy szemétkosár rajz, amibe a kijelölt adatokat „kidobhatjuk".
Hátránya ennek a parancsszervezésnek, hogy korlátozza a drive lehetőségeinek kihasználását. Nem tudjuk letörölni például a lemezen levő összes H betűs filet egy paranccsal, amelyet C 64 üzemben egyszerűen megtehetünk (print#15,"s0:H*"), kénytelenek vagyunk egyenként törölni minden egyes filet.
2. Egységes rendszerbe foglalták a különféle felhasználói programokat, ezek meghatározott felületen illeszkednek egymással. Ez alatt a felület alatt az általános adatszerkezetet, és programozási szintet értjük, amelyet a VLIR fileszervezés, és az ugrótábla jelent. Így például a rajzolóprogram rajzait felhasználhatjuk a szövegszerkesztőben, a táblázatkezelő adatait átvihetjük a szövegszerkesztőbe, a rajzolóprogramba és így tovább...
Egy ilyen egységes rendszer következményei szinte elképzelhetetlenek, sok felesleges munkát tudunk vele megtakarítani.
Az eddigi Commodore programok közül a Print Box az egyetlen, amely kevés sikerrel, de próbálkozik az egységesítéssel. Bár ez cseppet sem hasonlítható ehhez a szintű általánosítási törekvéshez.
3. A GEOS a fejlett programozási irányzat, a moduláris programozás jegyében készült. A modulok leírása, az egész rendszer felépítése jól dokumentált, könnyen fejleszthető. Nem új ez a módszer, már az UNIX-ki-fejlesztését is így csinálták, csak eddig kisgépen még nem találkoztunk hasonló megoldással. Általában a rendszerleírást késve adja ki a gyártó cég, esetleg más által visszafejtve jut a nyilvánosság elé. Az UNIX-nál kezdődött, hogy a rendszerleírás tartalmazza a forrásprogramot és a felhasználó ezt már úgy alakíthatja, ahogy akarja. A GEOS is hasonló, bár speciálisan egy processzorra (6510) készült. A rendszer leírása azonnal megjelent, nem titkolózták a programozói fogásokkal, a felhasználási lehetőségekkel.
4. Új és gyors file-kezelést hoztak létre a VLIR file-ok előállításával. A VLIR-file legnagyobb előnye, hogy az ember sokkal hosszabb programot írhat, mint a C 64-es tárkapacitása. Különálló részekre osztjuk fel a programot, melyeket azután a GEOS szükség esetén betölt. Ha például rákérdezünk egy C 64-es file-ra, és a GEOS azt az információt adja, hogy a file szekvenciális, akkor ez nem a megszokott értelemben vett soros elérésű adatfile, hanem csak arra utal, hogy szekvenciális tárfelhasználásról van szó. Ez olyan, mint ahogyan a BASIC-programok tárolódnak. A GEOS által használt VLIR-file-ok jelentése: „Variable Lenght Indexed Record" (változó hosszúságú indexelt rekordok)
A VLIR továbbfejlesztése a relatív tárkezelésnek. Egy file-t nem kell egy darabban betölteni, hanem több kisebb, különböző hosszúságú részletben. Előnye, hogy a felhasználó csak azokat a programrészeket tölti be a tárba, melyekre feltétlenül szüksége van. Jó példa erre a GEO-PAINT, illetve a vele előállított képek. Tehát egy VLIR-mutatóblokk a különböző programrészekre mutat, melyeket csak szükség esetén tölt be, és a VLIR file-ok nemcsak programok, hanem adatfile-ok esetén is használhatók. Pl. a Notesz-file minden VLIR-mutatója a notesz egy-egy lapjára mutat. A VLIR mutatóblokk max. 127 rekordra mutathat. (256 Byte hosszú lemezblokkot feltételezve).

A GEOS RUTINOK

Ha megismertük a GEOS mellé adott programokat (geoPaint, geoWrite) akkor vizsgáljuk meg, milyen rutinokból áll, és milyen a memóriafelosztás a GEOS alatt. A memóriafelosztás alapján láthatjuk, hogy a GEOS mellett csak gépi programok férnek el.
Ugyanis egy elég kicsiny memóriaterület áll a rendelkezésünkre ($0400–$5FFF). Ebben igazán jó programot csak úgy írhatunk, ha kihasználjuk a GEOS már bentlévő rutinjait.
A rutinokat funkciójuk szerint több nagy csoportba oszthatjuk:
- **Grafikus rutinok**
- **I/0 rutinok**
- **Belső rendszervezérlő rutinok**

Ezeket kisebb csoportokra bonthatjuk:
A grafikus rutinok:
– **alapvető grafikus feladatok:** pont kigyújtás, eloltás, vizsgálat, vonalhúzás, törlés, invertálás stb. A két képernyő közötti műveletek.
– **ablakkezelés:** ablak felépítése, szöveg kiírása az ablakba, a kiválasztható ablakocskák (OK, YES stb.) kirajzolása, lekérdezése.
– **menükezelés:** a vízszintes, ill. függőleges menük felépítése, lekezlése.
Az I/0 rutinok:
– **lemezműveletek** (direktory kezelése, validálás stb.)
– **file-műveletek** (törlés, átnevezés stb.)
A rendszervezérlő rutinok:
– **Hibavizsgálatok, üzenetek.**
– **Konfiguráció beállítása**

A rutinok vezérléséhez paraméterek kellenek. Ezeket a paramétereket a regiszterekben, de az utasítást (rutinhívás) követő táblázatban is átadhatjuk.

Mint tudjuk, a 6510-nek három regisztere van, és egy összetettebb funkció hívásához több mint három paraméter kell. Ezért a GEOS alatt létrehoztak 16 kiegészítő 16 bites regisztert a nullás lapon (S02–S22).

A másik nagy változás a GEOS-ban a lemezen tapasztalható. A normál DOS által üresen hagyott területeket használják ki a programok egyéb jellemzőinek (idő, dátum stb.) tárolására. Sajnos az egy programhoz tartozó információs blokk külön egy szektornyi helyet foglal, így kevesebb a felhasználható lemezterület. Ebben az információs szektorban van letárolva a filet jellemző ICON, töltési cím, az információs táblában található üzenetek.:

Memórifelosztás GEOS alatt:

$0000–$007F GEOS KERNAL által használt nullás lap
$0080–$00FF Az applikációk által használható nullás lap.
$0100–$01FF Processzor stack
$0200–$03FF Nagyrészt nem használt.
$0300–$0333-ig a normál C 64 rendszer vektorok találhatók itt. Ezeket a vektorokat a GEOS mindig meghatározott értékre állítja be.
$0400–$05FF Application programok számára fenntartott terület. Ha a DESK–TOP-hoz tartozó rutinokat is szeretnénk futtatni, használjuk a $4570–$496E területet.
$6000–$7F3F 2. Képernyő. Ide menti el a GEOS KERNAL az ablakok mögötti területet, vagy pedig a menük alatti részt.
$7F40–$7FFF Applikációk számára szabad hely.
$8000–$80FF Pufferterület a floppy adatoknak.
$8100–$81FF 2. Pufferterület (az aktuális infoszektort tartalmazza)
$8200–$82FF 3. Pufferterület (a GEOS által használt BAM)
$8300–$83FF 4. Pufferterület (a track-sector mutatókat tartalmazza töltés és tárolás esetén)
$8400–$8BFF GEOS KERNAL adatai (JOB címek, filenév, disk név, sprite adatok)
$8C00–$8FE9 Szín RAM
$8FF8–$8FFF A sprite adatok mutatói
$9000–$9FFD GEOS KERNAL (a program)
$A000–$BF3F 1. képernyő, az, amelyet a képernyőn látunk.
$BF40–$FFFF GEOS KERNAL

Honti Tamás

SZOFTVER ÖTLETEK



3 DIMENZIÓS TÖMBÖK ENTERPRISE-ON

Az Enterprise IS–BASIC-jének egyik hibája, hogy csak maximum kétdimenziós tömböket tud használni. Ez legtöbbször elég is, s ha nem, akkor sem túl bonyolult egy többdimenziós tömböt egy kétdimenziósba elpakolni (erre is mutat példát az alábbi program).

```
 10 INPUT PROMPT "TOMBMERETEK:":H1,H2,H3
 20 NUMERIC TOMB2(H1,(H2+1)*(H3+1)-1)
 30 DEF TOMB(I,J,K)
 40   LET TOMB=TOMB2(I,J*(H3+1)+K)
 50 END DEF
 60 DEF TOMBBE(I,J,K,TART)
 70   LET TOMB2(I,J*(H3+1)+K)=TART
 80 END DEF
 90 !
100 ! PELDA BEVITELRE, A TOMB(2,0,3)-BA
AKARUNK 6*78-AT VINNI:
110 CALL TOMBBE(2,0,3,6*78)
120 ! PELDA TOMBELEM KIOLVASASARA, A TOM
B(2,0,3) ERTEK FELERE VAGYUNK KIVANCSIAK
130 PRINT TOMB(2,0,3)/2
```

Ha valaki azonban ragaszkodik pl. a háromdimenziós tömbökhöz (szép, áttekinthető programot akar írni), annak segítségül közöljük ezt az egyszerű programot. A három indexhatárt a H1, H2, H3 változókba rakva a 20–80 sorok lehetővé teszik, hogy a TOMB nevű háromdimenziós tömböt kényelmesen használjuk – példát a használatra a 110-es és a 130-as sorokban láthatunk. A példában a tömb indexei 0-tól kezdődnek. Ezt a két eljárást bárki könnyen átalakíthatja 3-nál több dimenziós vagy string tömbökre, ill. úgy, hogy az indexek 1-től kezdődjenek. Az ilyen tömbök használata persze kicsit lassítja a programot, a TOMB egy elemének elérése pl. 2,2-szer lassabb, mint a TOMB2 megfelelő elemének az elérése.

Király Zoltán

KÖR LASER BASICBEN

Az LSI kiadásában nemrégiben megjelent a SINCLAIR SPECTRUM JÁTÉK ÉS PROGRAM III c. könyv. Ebben olvasható a LASER BASIC leírása. Szerencsémre 4–5 hónapja találkoztam, és mindenféle leírás nélkül kísérletezgettem vele. Szerintem is nagyon hatékony BASIC bővítés, főleg grafikai programok írásához. Egyetlen hibáját vettem észre, pontosabban a hozzá tartozó SPRITE GENERATOR-ét. Sok esetben jó lenne, ha lenne benne körrajzoló rutin is. Sajnos ezt viszont kifelejtették belőle, de könnyedén beírható. Mivel a program törzse BASIC, így BREAK-kel leállítható, majd az alábbi módosításokat kell elvégezni:

```
.
115 IF AC=94 THEN GOSUB 5300
.
.
5300 REM KöRRAJZOLAS
5310 LET KOZ=8*(XS+SX)+XC+CX: LET EPE=167-(8*(YS+SY)+YC+CY)
5320 INPUT "SUGAR = ";RAD
5330 IF KOZ+RAD>247 OR KOZ-RAD<128 OR EPE+RAD>167 OR EPE-
     RAD<48 THEN PRINT AT 21,0;"ERROR - KILOGNA A KEPBOL"
     : GO TO 5310
5340 LET INV=0: GO SUB 5400
5350 PRINT AT 21,0;"MEGFELEL?        Y/N ": GO SUB GTAC
5360 IF AC=89 THEN RETURN
5370 IF AC=78 THEN LET INV=1:
5400 CIRCLE INVERSE INV; KOZ,EPE,RAD: RETURN
```

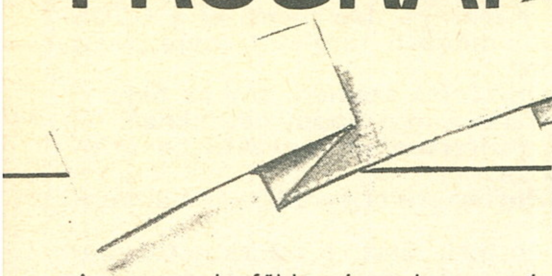
Tehát ennyi kis kiigazítással megoldható a körrajzolás. Először a középpontot kell a kívánt pozícióba állítani mind a spriteképernyőn, mind a munkatáblán. A rutin hívása a SYMBOL SHIFT H lenyomásával történik. Ezután kell megadni a sugár nagyságát, majd kirajzolódik a kör. Ha úgy sikerült, ahogy gondoltuk, akkor a kérdésre y-nal válaszolva térhetünk vissza a főprogramba. Ha netán nem úgy sikerült, akkor az n-nel válaszolva, az elhibázott kísérlet letörlődik és megjelenik egy újabb választási lehetőség, y lenyomásával újra próbálkozhatunk, n-nel visszatérünk a főprogramba.
A másik bővítési lehetőség Spectrum tulajdonosok számára. Az eredeti program a kurzor-nyilakra nem reagál, pedig sokkal kényelmesebb a használatuk. Ezek használata a következő módosításokkal érhető el:

```
.
110 IF AC=8 OR AC=9 OR AC=10 OR AC=11 OR AC=53 OR AC=54 OR
     AC=55 OR AC=56 OR AC=48 OR AC=57 OR AC=34 OR AC=44 THEN
  GO TO 1000
1015 IF AC=56 OR AC=9 THEN LET XC=XC+1: GO TO 1200
1025 IF AC=54 OR AC=10 THEN LET YC=YC+1: GO TO 1200
1030 IF AC=55 OR AC=11 THEN LET YC=YC-1: GO TO 1200
1035 IF AC=56 OR AC=8 THEN LET XC=XC-1: GO TO 1200
1045 IF AC=57 OR AC=44 THEN LET ON=65: GO TO 1210
1050 IF AC=48 OR AC=34 THEN LET ON=112: GO TO 1210
```

Németh Gábor

ötlet

PROGRAMAJÁNLAT:



GÉPI KÓDÚ PROGRAMOK DATA SOROKBAN – PRIMO

A program legfőbb erénye, hogy a gépi kódú program olyan BASIC-változatát készíti el, amely a legkisebb változtatás nélkül is azonnal indítható. A program ugyanis a betöltő programot is elkészíti, mégpedig úgy, hogy az az eredeti gépi kódú programmal azonos helyre tölt be, így tehát azzal egyenértékű működést biztosít. Ezen felül a program első soraként REM sort készít, ahova beírhatjuk kimentés előtt a program nevét.

Egyéb jellemzők, szolgáltatások:

– a gépi kódú program nagysága elvileg nem korlátozott
– a készítendő BASIC-program tárban való optimális elhelyezéséről a program gondoskodik
– a gépi kódú program kezdő- és végcímének ismeretében a DATA program lefutása után a rendszerváltozók automatikusan átírásra kerülnek úgy, hogy csak a létrehozott új program válik elérhetővé, amelyet most már szalagra lehet rögzíteni.
– indításkor az aktuális gépi kódú program kezdő és végcímének decimális értékeit kell beírni, ezután kiírásra kerül a művelethez szükséges időtartam.
– ha az aktuális gépi kódú program és a DATA program részben vagy egészben azonos területre kerül, akkor a program futása még a művelet megkezdése előtt üzenet kiírással leáll. A készített BASIC-program kimentése után a rendszerváltozók alaphelyzetbe állítását a gép rövid időre történő kikapcsolásával végezzük el.

A DATA program működése a következő:

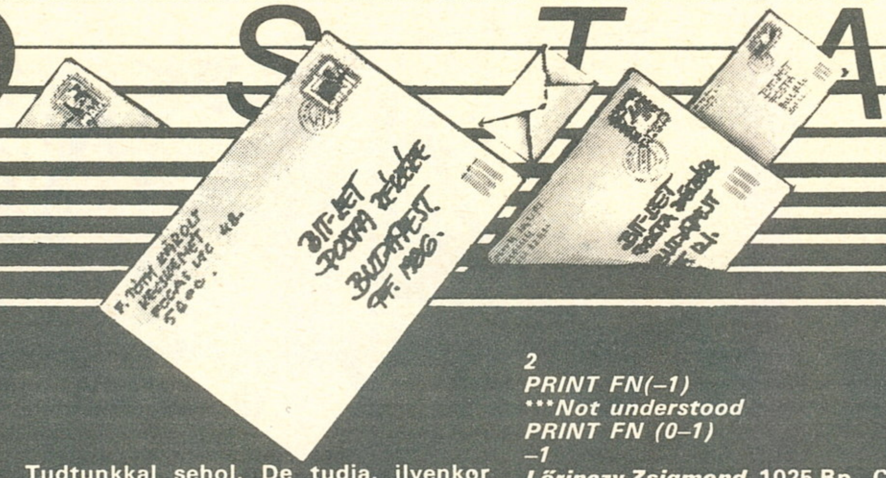
10–130 nem kíván magyarázatot
140–160 felmérik a működő és a készítendő programok tárigényét
170–180, 200–210 sorok a működő és a gépi kódú programtárban elfoglalt helyzetétől függően kijelölik a DATA file kezdőcímét
190 leállítja a program futását, ha a felmért tárigény és a gépi kódú program között átfedés jön létre
220 tapasztalati képlettel számolja és kiírja a művelet idejének közelítő értékét
240 az "1 DELETE1" sort készíti. Erre azért van szükség, mert a DATA program lefutása után a létrehozott BASIC-program magától indul.
250 a REM sort készíti el
260–410 A DATA sorokat készítik, ezenbelül a **350**-es sor a gépi kódú program végét figyeli
410–510 sorok készítik el a betöltőprogramot
520 átírja a rendszerváltozókat és JP 1B5D-vel NEW utasítás rutinjára lép. Ez indítja el a 43A4 címen kezdődő BASIC programot, melyről a **240**-es sornál esett szó, de a rendszerváltozók beállítása miatt erre szükség van.
540–570 a sorszámozáshoz készített szubrutin az **580**, ill **600**-as sorszámon kezdődő szubrutinok a V és K1 változók értékétől függően állítják be "P" értéket, amely adott esetben az emitett változók értékeit kettes komplemens kódra konvertálja.

Szekeres Antal Orosháza, Huba u. 4.

```
10 REM **********************************
20 REM *       Gépi kódu programok      *
30 REM *     DATA sorokká alakitása,    *
40 REM *   betöltö program készités     *
50 REM **********************************
60 REM * 1987.03.15. *  Szekeres Antal  *
70 REM **********************************
80 CLS
90 PRINTCHR$(2)" DATA FILE : BETOLTö "CHR$
(18)"            program készitése":PRINT:P
RINT:PRINT
100 REM változóknak helyfoglalás
110 Q=0:P=0:G1=0:G2=0:HE=0:BA=0:SZ=0:I=0:K
=0:V=0:K1=0:SO=0:A=-1:J=0:L=0:A$=" ":AB=0:
FB=0
120 INPUT"Gépi kódu program kezdöcime : ";
G1
130 INPUT"Gépi kódu program vége : ";G2:PR
INT:PRINT:PRINT
140 HE=(G2-G1)*4+(G2-G1)/8*6+170
150 BA=PEEK(16637)+256*PEEK(16638)
160 SZ=BA+HE
170 IF G1>SZ THEN K=BA:GOTO00220
180 IF PEEK(16548)+256*PEEK(16549)>G2THEN0
0200
190 CLS:PRINT"A gépikódu program azonos te
rületre esik  a BASIC programmal!":PRINT:P
RINT"Irja át a BASIC terület kezdöcimét és
 is- mételje meg a program betöltést.":PRI
NT:PRINT:PRINT:END
200 IF G1>17386+HE THEN K=17386:GOTO00220
210 K=PEEK(16637)+256*PEEK(16638)
220 PRINT"Türelem,kb.";INT(HE*.3/60)/10;"
percig dolgozom.":PRINT:PRINT
230 V=K
240 K1=V:V=V+2:SO=1:GOSUB00580:POKEV-P,SO-
256*INT(SO/256),INT(SO/256):V=V+2:GOSUB005
80:POKEV-P,182,49,0:V=V+3:GOSUB00600:POKEK
1-P,V-256*INT(V/256),INT(V/256):SO=0
250 GOSUB00540:POKEV-P,147,32,65,32,112,11
4,111,103,114,97,109,32,110,101,118,101,32
,58,0:V=V+19:GOSUB00600:POKEK1-P,V-256*INT
(V/256),INT(V/256)
260 K1=V:V=V+2:GOSUB00580
270 SO=SO+10
280 POKEV-P,SO-256*INT(SO/256),INT(SO/256)
,136,32:V=V+4
290 FORI=1TO8
300 A=A+1:IFG1+A>32767THENQ=65536ELSEQ=0
310 A$=STR$(PEEK(G1+A-Q)):L=LEN(A$)
320 FORJ=2TOL:GOSUB00580
330 POKEV-P,ASC(MID$(A$,J,1)):V=V+1
340 NEXT:GOSUB00580
350 IF G1+A=G2 THEN POKEV-P,0:V=V+1:GOSUB0
0600:POKEK1-P,V-256*INT(V/256),INT(V/256):
GOTO00410
360 IF I<8THENPOKEV-P,44:V=V+1
370 NEXT:GOSUB00580
380 POKEV-P,0:V=V+1:GOSUB00600
390 POKEK1-P,V-256*INT(V/256),INT(V/256)
400 GOTO00260
410 GOSUB00540
420 POKEV-P,71,49,213:V=V+3:GOSUB00580:A$=
STR$(G1):L=LEN(A$):FORI=2TOL:POKEV-P,ASC(M
ID$(A$,I,1)):V=V+1:GOSUB00580:NEXT:POKEV-P
,58:V=V+1:GOSUB00580
430 POKEV-P,71,50,213:V=V+3:GOSUB00580:A$=
STR$(G2):L=LEN(A$):FORI=2TOL:POKEV-P,ASC(M
ID$(A$,I,1)):V=V+1:GOSUB00580:NEXT:POKEV-P
,58,143,71,49,212,51,50,55,54,55,202,81,21
3,54,53,53,51,54,58,149,81,213,48,0
440 V=V+24:GOSUB00600:POKEK1-P,V-256*INT(V
/256),INT(V/256)
450 GOSUB00540
460 POKEV-P,129,73,213,71,49,189,71,50,58,
143,73,214,51,50,55,54,56,202,74,213,73,58
,149,74,213,73,206,54,53,53,51,54,0:V=V+33
:GOSUB00600:POKEK1-P,V-256*INT(V/256),INT(
V/256)
470 GOSUB00540
480 POKEV-P,139,65,58,177,74,44,65,0:V=V+8
:GOSUB00600:POKEK1-P,V-256*INT(V/256),INT(
V/256)
490 GOSUB00540
500 POKEV-P,135,73,58,65,213,193,40,71,49,
206,81,41,0:V=V+13:GOSUB00600:POKEK1-P,V-2
56*INT(V/256),INT(V/256)
510 GOSUB00580
520 POKEV-P,0,0:GOSUB00600:POKEK1-P,V-256*
INT(V/256),INT(V/256):POKE16548,K-256*INT(
K/256),INT(K/256):V=V+2:POKE16722,V-256*IN
T(V/256),INT(V/256):POKE16755,42,82,65,34,
249,64,195,93,27:CMD
530 END
540 K1=V:V=V+2:GOSUB00580
550 SO=SO+10
560 POKEV-P,SO-256*INT(SO/256),INT(SO/256)
:V=V+2:GOSUB00580
570 RETURN
580 IFV>32767THENP=65536ELSEP=0
590 RETURN
600 IFK1>32767THENP=65536ELSEP=0
610 RETURN
```

CSERE-BERE

POSTA



A Primoknál, az "A" típusnál ('84 1. verzió) mint köztudomású, nincs kivezetve semmiféle csatlakozó (kivéve a szükségeseket). A billentyűzetet pedig nem éppen a játékprogramokhoz tervezték, s így előbb-utóbb szükségessé válik a botkormány.
Mondjuk üggyel-bajjal összehozok valamiféle megoldást. Akkor vajon bármilyen joystick csatlakoztatható lesz hozzá? (a hardver füzetben ez áll: „megfelelően megépített")
Kovács Zoltán, 1196 Bp., Jánh F. u.147.

A teljes kiépítésű Primon a joystick részére rendelkezésre áll az egyik kimeneti regiszter egyik bitje és az egyik bemeneti regiszter két (azaz kettő) bitje. Ebből mindjárt kitűnik, hogy nemcsak szoftverre, hanem speciális hardverre (illesztőegység, "interface") is szükség van. Vagyis lenne. Az ős-Primokból ugyanis hiányzik néhány IC, a csatlakozókról nem is beszélve.
Elkeseredésre azonban nincs ok:
1. A µMagazinban (1986. október) közölték a botkormány interface kapcsolási rajzát és a hozzátartozó gépi szubrutint. (31. oldal, Gerencsér Sándor)
2. µMagazin 1985/2, 28. oldal: dr. Simonyi Endre vállalja, hogy 50 Ft-ért bármilyen gyári botkormányt átalakít olyanná, hogy Primohoz is lehessen használni. (Gondolom, akkori 50 Ft-ért.) ...Vajon a hiányzó IC-ket is beszereli?
3. Érdeklődésünkre a Fehér Hajó utcai GELKA-ban azt mondták:
a) Vállalják a Primo hiányzó integrált áramköreinek beszerelését a csatlakozókkal együtt.
b) Készítenek joystickhez illesztőegységet (kívánság szerint egy vagy két botkormányhoz), a számítógép dobozába beszerelik, és így Commodore joysticket lehet hozzá használni. Árakat nem mondtak, éppen most változnak...

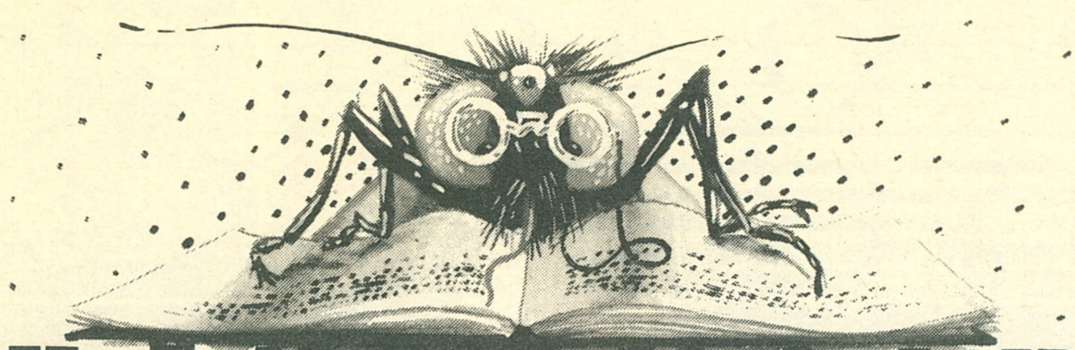
Darázs Tibor, Debrecen, István út 19. Primo tulajdonos olvasónk egy sor kérdést tett föl. Igyekeztünk válaszolni.
1. Lehetőség van-e P64 tip. gépbe a soros interface kialakításához szükséges elemek házilagos beépítésére, és egy ilyen átalakítással alkalmassá válik-e a gép a Commodore-lemezmeghajtó kezelésére?
Igen, a Hardver-füzet 1. és 2. sz. mellékletei alapján minden további nélkül.
Ismét igen.
2. Értesülésem szerint kapható kazettán a lemezegység kezelési feladatait ellátó (CDOS című) operációs rendszer. Hol kapható és mennyiért?
Láttuk árusítani a Keravillnál, az Elektromodul árjegyzékén a kiskereskedelmi ára 1986-ban 1103 Ft volt.
3. Hol szerezhető be a Commodore lemezmeghajtó soros csatlakozójának ellendarabja?
Tudtunkkal sehol. De tudja, ilyenkor mindig azt szoktuk tanácsolni, hogy próbálkozzon a Bécsi út túlsó végén, ott sem biztos hogy sikerül.
4. A PRIMO Szoftverleírás szerint a 00161-es címen található a képkioltásra várakozó (VBLANK) rutin, amely a képkioltás kezdetekor adja vissza a vezérlést a programnak. Azonban: az Y= =call (161) módon meghívva, a gép alapállapotba kerül és a program elszáll. Mi az oka? A cím nem 161?
A képkioltásra várakozó rutin a 12449 (30A1H) címen kezdődik, bár a "Szoftverfüzet" mást állít.
BASIC-ben egyáltalán nincs értelme meghívni, gépi kódú programoknál is csak akkor, ha valami hiperszuperprofi programot akarunk csinálni.
5. A képkioltás 7 ms-nyi ideje alatt hogyan lehet gyorsan megváltoztatni a képernyő tartalmát?
A képernyőhöz tartozó memóriaterület minden Primonál az utolsó 6K. (A kimeneti regiszter módosításával ezt 8K-val alacsonyabb címre lehet átkapcsolni.) A képkioltás ideje alatt a gyors gépi kódú rutinokkal erre a területre tetszőleges adatokat lehet írni (betölteni).
Ha a képernyőmódosítást megfelelően szervezzük meg, a VBLANK rutint – tapasztalatom szerint – fölösleges alkalmazni, a kép gyors módosítását a szem úgy érzékeli, mintha az egyszerre történne – éppen úgy, mint egy filmen.
Példa a célszerű szervezésre: a számításokat és a képmódosításokat a RAM-nak egy másik területén végezzetjük el és a kész képet egy ciklussal (sőt! egy utasítással [!]) átvisszük a képernyőmemóriába.
Egy másik jó módszer az, ha OUT utasítással kapcsolgatjuk a két képernyőt, és mindig éppen arra a képernyőre rajzolunk, amelyik nem látszik. Ez utóbbi esetben jól megvagyunk mindenféle gépi szubrutinok nélkül is.

Kérem továbbítsák kérdéseimet a Videoton illetékeseinek; és közöljék, hogy a TV-Computer leírásában miért "titkolták el":
*– a ***BASIC corrupted hibaüzenetet,*
– az IO függvényt (argumentuma 0 és 255 kötött kell legyen, értéke általában nulla, 87 és 106 közötti értékre viszont különböző számokat ad ismeretlen "logika" szerint), (tokenje sincs)
– azt a tényt, hogy a DEF F(X) ... utasítással definiált függvények argumentuma nem kezdődhet "–" jellel pl.:
10DEF FN(X) = X
RUN
PRINT FN(2)
2
PRINT FN(–1)
****Not understood*
PRINT FN (0–1)
–1
Lőrinczy Zsigmond, 1025 Bp., Cimbalom u. 1.
Kérdéseire nem tudunk válaszolni. Tartunk tőle, hogy az illetékesek sem tudnának. De örülünk az információnak.

Szilágyi Balázs, 1125 Bp., Nógrádi u. 2/B olvasónk több kérdéssel fordult hozzánk, amelyekkel megkerestük Gergely Jánost a Videoton Vevőszolgálatán, akitől a következő felvilágosítást kaptuk.
1. Létezik-e még Intelligent Software nevű cég? Ha van, mi a címe?
A cég létezik, de senki (a céggel szoros kapcsolatban álló VIDEOTON-t is beleértve) nem hajlandó címüket kiadni. (Magyarázatot sem kaptunk.)
2. Milyen jogok védettek a VIDEOTON, az Enterprise GMBH vagy az IS Enterprise, vagy TV-Computer
– ROM listájának közreadása (saját megjegyzéseimmel)
– kapcsolási rajz publikálása
– egyéb szoftver-, ill. hardverismeretek nyomtatása
A TVC-ről tudunk felvilágosítást szerezni:
– ROM listát a VIDEOTON nem publikálhatja, de annak nincs akadálya, hogy más, a saját megjegyzéseivel ezt megtegye.
– Kapcsolási rajz publikálása folyamatban; van a Novotrade-ben készül egy 3 darabból álló sorozat:
- Operációs rendszer ismertetése
- Hardverismertetés (kapcs. rajzzal)
- BASIC példatár

Időpontot nem tudtak mondani, de minden könyvesboltban kapható lesz.
3. Ha van angol és német billentyűzettel ellátott Enterprise, akkor miért nem csinálnak olyat, amit a TV-Computeréhez hasonló dobozba tesznek? (Úgy jobban vonzana.)
Nem világos, miért kellene az Enterprise-t TVC dobozba tenni?
4. Miért nem olvashatók a BIT-LET-ben statisztikák a hazai mikroszámítógép-tulajdonosokról? Hányan vannak? Melyik gép a legnépszerűbb?
Mert erre csak becslések vannak, durván 100%-os hibával, és így az adatok nem sokat mondanak. (Azt meg úgyis mindenki tudja, hogy C 64-ből van a legtöbb).
5. A TVC drive leírása 250 példányban készült. Ezután érdekel, hogy vajon hányan vannak az Enterprise és a TV-Computer tulajdonosok.
A TVC drive leírás nem 250, hanem 2500 példányban készült legalább. Minden drive-hoz jár. Így azt hisszük, hogy elegendő.

ötlet

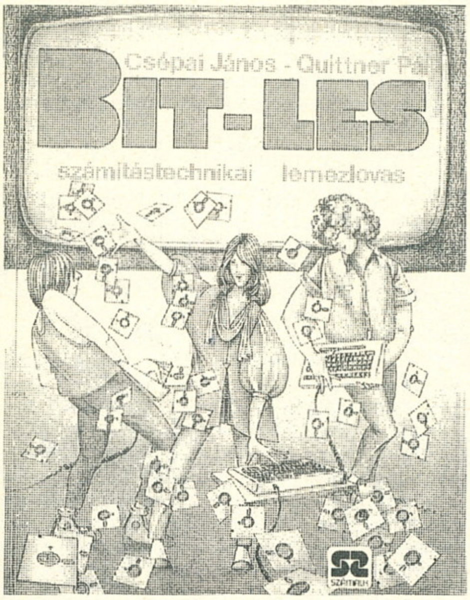


KÖNYVMOLY

Peter Norton: Fedezzük fel az IBM PC-t! – Műszaki Könyvkiadó, 349 o., 110 Ft. (Az IBM-könyvek és -programok neves szerzője e kötetében saját, közvetlen tapasztalatai alapján írja le a gép megismerésének folyamatát.)

Theisz György: BASIC-tanácsadó C-16, Plus/4
(A didaktikusan felépített kötet a gépek iskolai felhasználóinak – tanároknak és diákoknak – nyújt segítséget.)

Csépai János–Quittner Pál: BIT-LES – Számítástechnikai lemezlovas – SZÁMALK, 223. o., 125. Ft.
Ha lapunk rendszeres olvasói egy könyvesboltban találkoznak ezzel a kötettel, bizonyára felmerül bennük a kérdés: „Mi ez? Talán valami újfajta SZUPER BIT-LET?"
Már elöljáróban meg kell jegyeznünk, hogy a könyvnek és a BIT-LET-nek semmi köze nincs egymáshoz. A szerzők és a kötetet kiadó SZAMALK arra hivatkoznak, hogy a címet a Beatles együttes nevéből asszociálva ötlötték ki. Erre utalna a „Számítástechnikai lemezlovas" alcím, a számítógépen zenélő fiatalokat ábrázoló címkép, valamint a belső címlap a Sárga tengeralattjáró c. film figurájával. Szerintük így a cím ötlete mindenki számára nyilvánvaló – de erre konkrét utalást a kötetnek szinte a legvégén, a 201. oldalon tesznek csupán. Ahogy elmondták, csak két héttel a könyvcím kitalálása után jutott eszükbe, hogy igencsak nagy az egybeesés lapunk címével.
Már csak a fentiek miatt is érdemes megnéznünk, hogy milyen színvonalú a könyv, hogy mennyiben szól lapunk olvasóközönségéhez, egyszóval vállalható-e számunkra a cím hasonlósága. A szerzők a bevezetésben 12 és 99 év között bárkinek ajánlják kötetüket. Ahogy az előszóból („A szerzők mentegetőzése") kiderül, a könyvet azért osztották 69, néhány oldalas fejezetre, és azért választották a könnyed hangvételt, hogy bárki kedve szerint lapozgathasson benne, illetve bármikor könnyen megtalálja az őt érdeklő témakör rövid ismertetését.
Ennek a szerkesztési módszernek azonban két nagy hátránya is van. Most mindkettőre csak egy-egy példát említünk – de hasonlóak, sajnos, hemzsegnek a kötetben.
Az egyik az, hogy a 69 kis fejezet mindegyike a számítástechnika egy-egy önálló témakörét igyekszik bemutatni a rendelkezésre álló két-három oldalon – ami szinte megoldhatatlan feladat. A szerzők sem képesek erre, így azután érvényesül a „sokat markol, keveset fog" elve: szinte mindenről szólnak, de nem mondanak szinte semmit. (Sajnos, igaz, amit a szerzők a 153. oldalon önironikusan megjegyeznek: „....önmagában attól nem válnak valakik íróvá, számítástechni-



kusokká, vagy egyidejűleg mindkettővé, mert megírtak egy számítástechnikával foglalkozó könyvet.") Így a mindössze másfél oldalas 11. fejezet úgy indul, mintha meg akarná magyarázni a bináris számok fogalmát és használatuk módját, azután egy váratlan fordulattal a Boole-algebra bemutatására tér rá, végül egy furcsa átvezetés után a gépi kód féloldalas ismertetésével zárul.
Van azért persze öt nagyobb szerkezeti egysége is a könyvnek: az első általánosságban igyekszik ismertetni a számítógép és a számítástechnika fogalmát, a második a hardverrel, a harmadik a szoftverrel, a negyedik a programozással, az ötödik pedig a számítástechnika alkalmazásával foglalkozik. A gond itt az – és ez a második, amit a szerkesztési koncepció kapcsán meg kell említenünk –, hogy az öt rész között rengeteg az átfedés, illetve, hogy az egymáshoz kapcsolódó témakörök nem követik logikus rendben egymást. Az átfedésekre ugyan a szerzők már az Előszóban felhívják a figyelmet. A fejezetek sorrendjében azonban már elhibázottnak tűnik az, hogy például a számítógépes rendszerek biztonságáról, az adatvédelemről szólnak a 39. és a 40. fejezetben – ahol sci-fi filmből és a valóságból vett eseteket mutatnak be az illegális adatfelhasználásra, és az adatvédelmi eljárásokat ismertetik –, valamint a 63. és 65. fejezetben, ahol a rejtjelezéssel, kódolással, és a katasztrófák elhárításával foglalkoznak. Sokkal közérhetőbb és használhatóbb lett volna, ha ez a négy fejezet egymás mellé kerül.
E kifogások mellett még azt is meg kell említenünk, hogy az öt fő rész színvonala, illetve ismereteik szintje sem egyenletes. Az első a laikusoknak szól. A második is így indul, de itt már sok az olyan fogalom, amit az amatőrök meg sem értenek. A harmadik rész szinte csak olyan fogalmakkal foglalkozik, amivel az otthoni felhasználó nem is találkozik – a szakmabeli számára viszont túl leegyszerűsítőek a magyarázatok. A negyedik és ötödik rész ismét csak a kezdőkhöz szólna – de rendszertelenül kapkod a témakörök között, szinte esetlegesen válogatva.
A könyv utolsó előtti – 68. – fejezete terjedelmesebb, mint a többi: „Nagy nevek, kis történelem" címmel a számítástechnika kiemelkedő egyéniségeinek munkásságáról olvashatunk 11 oldalon. Ez tulajdonképpen már a függelék részének tekinthető, hiszen a 69. fejezet rövid epilógusa után mellékletként egy tárgymutatót, valamint egy számítástechnikai kisszótárt is találunk a kötetben.
Külön kell még szólnunk a kötet grafikáiról, melyeket Halász Géza készített. A karikatúrákat – ahogy az 54.

fejezetben ő maga írja – az Art Studio Spectrumos változatával rajzolta. Nem lehetett kis munka, hiszen a kötetben három–négy oldalanként találunk illusztrációt. Időnként zavaró, hogy a rajzokban szinte minden zárt terület fel van töltve valamilyen rasztermintával – de talán a grafika a kötet legszínvonalasabb része.
Hogy válaszoljunk a kritika elején föltett kérdésre: nem örülünk, hogy a könyv ezzel a címmel látott napvilágot!

Tallér József

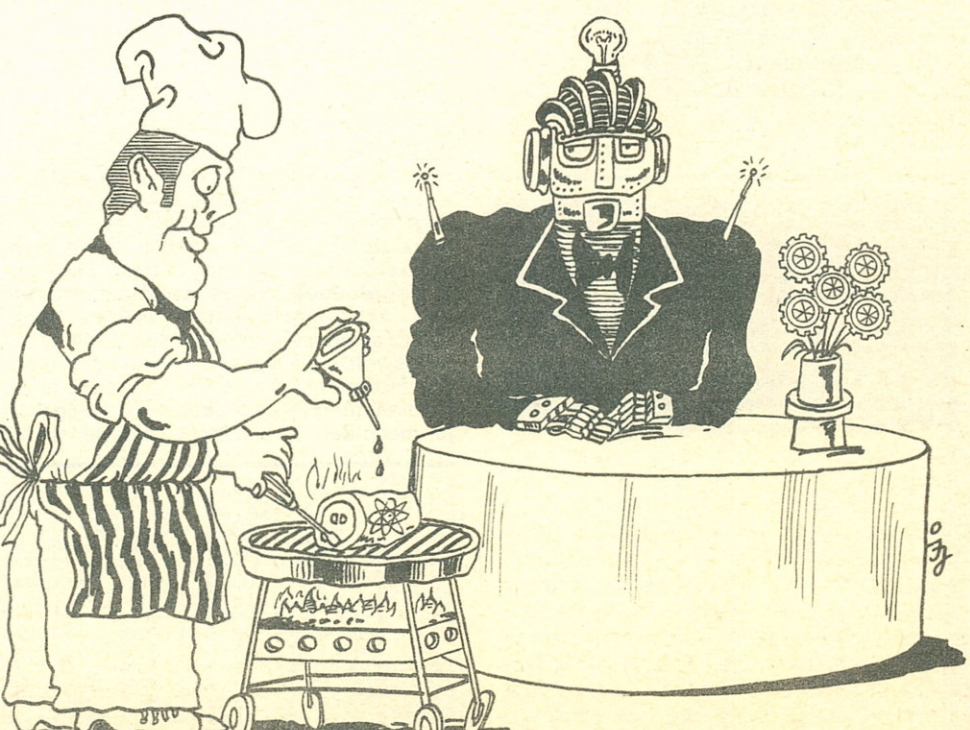
A. Hettinger–A. Heinz: ATARI-BASIC kezdőknek – Műszaki Könyvkiadó, 130 o., 60 Ft.

Nagy örömmel vettem észre a könyvesbolt polcán A. Hettinger–A. Heinz: ATARI-BASIC kezdőknek c. könyvét. Végre! Úgy tűnt, már annyira „leírták" az ATARI 400-as és 800-as típusait, hogy semmilyen irodalmat nem adnak ki róla, hozzá. Amikor azonban kézbe vettem a könyvet, kicsit meglepődtem: egy kezdőknek szóló könyvet vastagabbnak képzeltem. A tartalomjegyzéket szemlélve már erősen csodálkoztam: az ATARI-BASIC-et egyetlen fejezet (24 oldal) ismerteti. Ugyanakkor (ez előnyös) a grafikát és a hanggenerátor használatát külön fejezetben tárgyalja. Sajnos erős a gyanúm, hogy egy kezdő, aki ebből a könyvből akarja a programozást megtanulni, nehezen boldogul.
Mindenekelőtt túl rövidnek és felületesnek érzem a magyarázatokat. A program lényegét például így foglalják össze a szerzők: „Nem jó, ha a számítógép minden parancsot azonnal teljesít" (13. o.) Persze értem, de biztos, hogy ez egy kezdő számára is világos? A magyar fordítás lektora, Fried Katalin ugyan sok hasznos kiegészítő lábjegyzettel próbálta a könyv hiányosságait pótolni, ám azért a kiegészítések mégsem vehetik át a könyv szerepét. Pl. a lábjegyzetből kissé bővebben értesülünk a program mibenlétéről, itt van leírva először a sorok törlése (14. o.), kizárólag itt kapunk képet a változók fogalmáról (12. o.), a szerzők ugyanis rögtön a tömbök ismertetésével kezdik. Nem szerepel azonban – még lábjegyzetben sem – a változónév helyesírása (betűvel kezdődő betűszám kombináció) stb.
Elvi tévedések is előfordulnak. Csak néhány példa:
„A sorok számozása egytől ötjegyű számokig terjedhet..." – 32767-nél nagyobb sorszám nem írható!
„... a program futását a BREAK vagy SYSTEM-RESET billentyű lenyomásával lehet megszakítani. A megszakított programokat a CONT paranccsal lehet folytatni." – SYSTEM-RESET után a CONT nem működik (pontosabban a program **második** sorára ugrik)! (29. o.)
„Írjon egy programot, amely a következő színregisztereket állítja be: Keret: zöld, Háttér: vörös, Szöveg: fekete" – 0. grafikai módban (melyről itt szó



van) a háttér és a szöveg csak azonos színű lehet! (38. o.)
A 128. oldal utolsó bekezdése teljes tévedés. A GET, ill. PUT egy byte-ot mozgat, az INPUT sem ötöt, hanem annyit, ahány karakter hosszú az adat. A gép egyébként a számokat nem öt, hanem hat byte-on tárolja.
A helyzetet bonyolítják a didaktikai hibák, elírások, rendszertelenségek. Például: a PRINT ugyan külön alcímként szerepel, a PRINT-vessző és -pontosvessző használata azonban a STEP ill. az INPUT ismertetésekor van leírva – ki tudja, miért?
Az ‚A PRINT-parancs speciális alkalmazásai' címszó példái cseppet sem speciálisak az addigiakhoz képest (pl. ?„HALLO") 12. o.
A 38. oldal felső táblázata (3.1. tábl. Grafikus módok színregiszterei), az előtér (szöveg) színregisztere' teljesen összevissza és hibás.
A 44. oldal ábrája hibás és zavaró.
A 71. oldal az 5. fejezet kezdete: BASIC-parancsok. Az első négy „parancs": ABS, ADR, ASC, ATN. Ezek függvények! Parancsként beírva őket, hibajelzést kapunk. Ezenkívül az ADR függvény (a szöveggel ellentétben) csak alfanumerikus változóval használható. A BYE parancs leírásánál nem értem, mi történik automatikusan.
84. oldal: "A hang elhallgattatásának felélesztése". Tessék?
Végül bizonyos dolgokat meg sem említ a könyv: az Editálásról, vagyis sorok beírásáról, javításáról, a vezérlőbillentyűk szerepéről; a relációs jelekről szó sem esik.
Sokáig folytathatnám a hibák felsorolását.
El kell ismernem, hogy a magyar fordítás lektori megjegyzései sokat javítanak és segítenek. Nem volt szerencsés éppen ezt a könyvet lefordítani. Ráadásul a magyar kiadáshoz fűzött Kiegészítésben is vannak hibák.
Összefoglalva az a benyomásom, hogy ez a könyv egy rosszul megválasztott olvasókörnek készült. Kezdőknek túl rövid és érthetetlen egy ilyen „gyorstalpaló" tanfolyam. A haladóbbak viszont nem fognak bepötyögni olyan példaprogramokat, amelyeknek fele végrehajtásra sem kerül. Legvalószínűbb, hogy egyszerűen átlapozzák az egészet és a játékprogramokat fogják beírni a gépbe.
Hadd idézzek egy (nem tőlem származó) kritikát: „Nagy előnye, hogy magyarul van"! Hát, igaz...

Rieth József



tapasztalat

NYERŐ

ÉRTÉKELÉS

A pályázatra (bűvös kocka) 20 megoldás érkezett, melyek közül hatban a beküldött programrészlet durva hibákat tartalmaz. A többi 14 megoldás lényegében jó, van néhány nagyon ötletes elgondolás, néhányan viszont jó ötletüket nagyon csúnya programmal valósították meg. Ezért most is súlyozott sorsolást tartottunk(!) a BIT-LET Karácsonyon. Mivel erre lapzárta után került sor, a nyertesek nevét még nem tudjuk. **A sorsolás rendszere a következő volt:**

Az 1. díjat (3000 Ft-os Centrum-utalvány)
Bolgár László, Bp; Lévai Gábor, Albertirsa és Lipi Gábor, Bp. között,

A 2. díjat (1000 Ft-os Centrum-utalvány)
Blaschekk József, Bp.; Kruzslicz Ferenc, Tótkomlós; Szalkai István, Devecser és Szirmai Ákos, Bp. között sorsoltuk ki.

A 3–7. díjak, egyenként 1–1 db 200 Ft-os Centrum-utalvány) sorsolásán a fentieken kívül részt vehettek még:
Csillag Péter, Bp.; Csizmadia Tamás, Bp.; Hernyik András, Szentes; Kajári Gábor, Cegléd; Nagy Zoltán, Székesfehérvár; Peták Tamás, Szolnok; és Tőkés Mihály, Dunakeszi.

A feladat az volt, hogy olyan struktúrát találjunk ki a bűvös kocka állapotainak tárolásához, mely egy esetleges visszaforgató-programhoz a lehető legjobban illeszkedik; s meg kellett írni az egy oldal elforgatását megvalósító rutint.

Néhány észrevétel:

– mivel általában legfeljebb 72 elemszámú tömbökről van szó, ezért nem nagyon érdemes a memóriával takarékoskodni.

– egy kocka-visszaforgató algoritmusban úgyis elég sokat kell számolni, így egy forgatás megvalósítását nem érdemes a lehető leggyorsabbra készíteni, persze azért túl lassúra sem.

– egy oldal forgatásánál (bármilyen tárolásnál) 4-es ciklikus cseréket kell végezni, ezt nem érdemes FOR ciklussal csinálni. Főleg nem érdemes erre egy másik tömböt használni, s onnan visszamásolgatni. Sokkal egyszerűbb (és mellékesen gyorsabb is) a 4-es cserét az 5 értékadás leírásával megoldani.

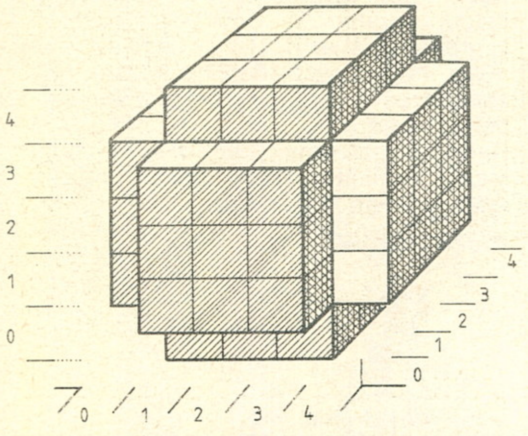
– mivel minden visszaforgató algoritmusban nagy szerepe van a szemközti oldalpároknak, így a szemközti oldalak kódjai között legyen egyszerű összefüggés (pl. dobókockaelv).

A megoldások 2 fő csoportba oszthatók. Volt, aki a kocka térbeli helyzetét tárolta (tehát az egyes kis kockák helyét és helyzetét), s volt, aki a lapokon látható négyzeteket. Egy visszaforgató algoritmus számára az első módszer tűnik jobbnak, azonban itt elég nehéz jól megoldani a kis kockák helyzetének tárolását. Erre több ügyes megoldás érkezett, a legizgalmasabbat alább közöljük.

A kis négyzetek tárolásánál az a gond, hogy egy forgatás a 4 szomszédos oldal 3–3, a forgatandó lap felé eső kis négyzetét is mozgatja. Ezeket legtöbben nagy segédtömbökkel és sok DATA-val határozzák meg, néhányan azonban ügyesebbek voltak. A megoldók e csoportjában Szirmai Ákos volt a legleleményesebb. ő a kocka lapjainak kis négyzeteit úgy számozta be, hogy azok a kocka egy-egy lapján csavarvonalat alkotnak. Sikerült elérnie azt is, hogy a kocka bármely oldalát forgatva, az adatok hasonlóképpen rendeződnek minden lapon. Azaz ha a kocka bármelyik lapját fordítjuk el, a további négy kapcsolódó oldalon a mozgatandó kis négyzetek sorszámai előre meghatározottak, a kiválasztott (forgatandó) oldal sorszámától függetlenek. Mindig a szükséges 3 db 4-es ciklikus cserét hajtja végre a gép, azaz megfelelő helyekre az új színek kerülnek. Ekkor csak az egyes lapok 4 szomszédjának kódját kell megfelelő sorrendben DATA-ba írni. Egy másik igen jónak látszó ötlet Bolgár Lászlótól származik, aki az egyes helyeken nemcsak az illető négyzet színét tárolja, hanem annak a helynek a sorszámát is, ahová rendezés után az illető lapon kerül.

A legötletesebb megoldás Lipi Gábor és Nyilas István László közös műve, levelükből és programjukból a lényeget kiragadtuk, s alábbiakban idézzük. Programjukból csak a forgatást végző rutint emeltük ki, s az egyszerűbb nyomtatás kedvéért „átcommodoreplusnégyesítettük", (az eredeti TVC-re szólt), bár ezzel nem sokat változtattunk rajta.

A Bűvös kocka állapotának tárolására egy 5*5*5-ös tömböt használtunk. Azért esett erre a választásunk, mert így egyszerre lehet nyilvántartani a 27 kis kocka egymáshoz viszonyított helyzetét és az 54 színes négyzet laponkénti helyét. Az előbbi segítséget nyújthat egy olyan rendezőprogramnak, amely a felcserélődött kockákat viszi a helyükre; az utóbbi pedig a már helyén lévő kockák megfelelő irányba fordításához nyújt segítséget. Így egyben a kocka megrajzolását is könnyű végrehajtani és egy későbbi mintázatfelismerő programrész is nehézség nélkül kezelheti az állományt.



Az ábrán az általunk használt háromdimenziós tömb térbeli szerkezete látható:
A be nem jelölt részeket nem használjuk. Ezen az ábrán az oldalaknak megfelelő rekeszeket láthatjuk, ezekben tároljuk a különböző színeknek megfelelő kódokat 0–5-ig.
Az ábrán látható testnek a belseje nem üres. Itt foglal helyet az a 3×3×3-as rész, amiben a kis kockák sorszámát tároljuk 0–26-ig.
A forgatást úgy hajtjuk végre, mint a valóságban.
A program forgatás szubrutinja a megfelelő laphoz tartozó két réteget a tömbben 90°-kal elforgatja.

És végül zárszóként: a Nyerő oldal azzal búcsúzik olvasóitól, pályázóitól, hogy reméltetőleg az új BIT-LET-ben is lesznek pályázatok, s lesznek kisorolható gépek.

```
10 DIM K(4,4,4)
50 INPUT "FORGATANDO LAP SORSZAMA";L
60 INPUT "FORGATAS MERTEKE (1-3)";MA
70 IF L AND 1 THEN MA=4-MA
100 FOR MB=1 TO MA:GOSUB 1000:NEXTMB:GOTO 50
1000 S=(1 AND L)*3:ON INT(L/2)+1 GOTO 1030,1040,1050
1030 FOR SK=S TO S+1:FOR I=-(SK=0)-(SK=4) TO 1:FOR J=I TO 3-I
1032 Z=K(SK,I,J):K(SK,I,J)=K(SK,4-J,I):K(SK,4-J,I)=K(SK,4-I,4-J)
1034 K(SK,4-I,4-J)=K(SK,J,4-I):K(SK,J,4-I)=Z:NEXT J,I,SK:RETURN
1040 FOR SK=S TO S+1:FOR I=-(SK=0)-(SK=4) TO 1:FOR J=I TO 3-I
1042 Z=K(I,J,SK):K(I,J,SK)=K(4-J,I,SK):K(4-J,I,SK)=K(4-I,4-J,SK)
1044 K(4-I,4-J,SK)=K(J,4-I,SK):K(J,4-I,SK)=Z:NEXT J,I,SK:RETURN
1050 FOR SK=S TO S+1:FOR I=-(SK=0)-(SK=4) TO 1:FOR J=I TO 3-I
1052 Z=K(I,SK,J):K(I,SK,J)=K(J,SK,4-I):K(J,SK,4-I)=K(4-I,SK,4-J)
1054 K(4-I,SK,4-J)=K(4-J,SK,I):K(4-J,SK,I)=Z:NEXT J,I,SK:RETURN
```

BIT-LET Szerkesztőségünk címe: ÖTLET Budapest 1986. Szerkesztő: Angyalosi László. Művészeti szerkesztő: Pribelszky Pál.
Tördelő: Domokos Imre. Hírrovat: Csányi György. Gépnyerő: Király Zoltán. Posta: Halász Péter.